

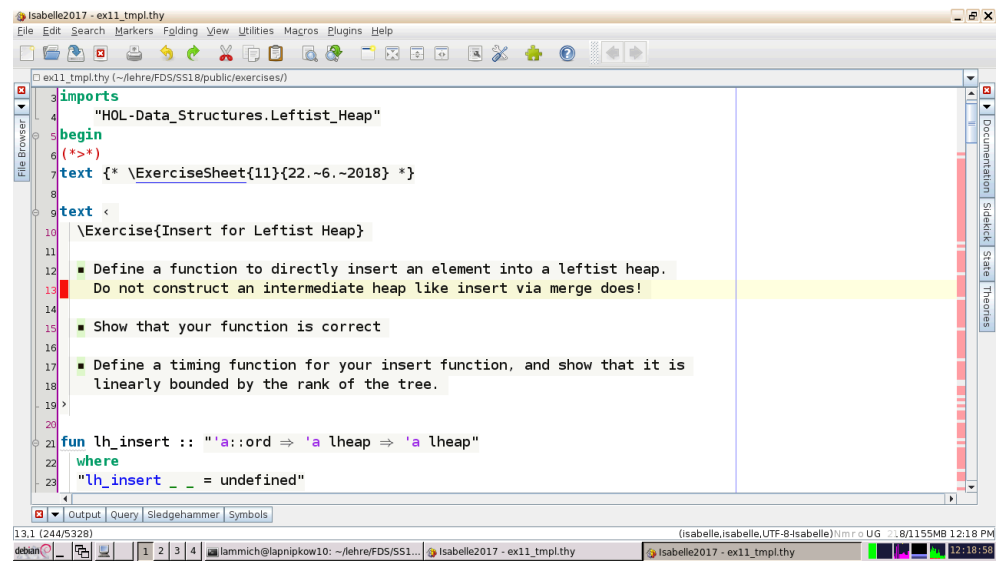
Script generated by TTT

Title: Lammich: FDS Tutorial (22.06.2018)

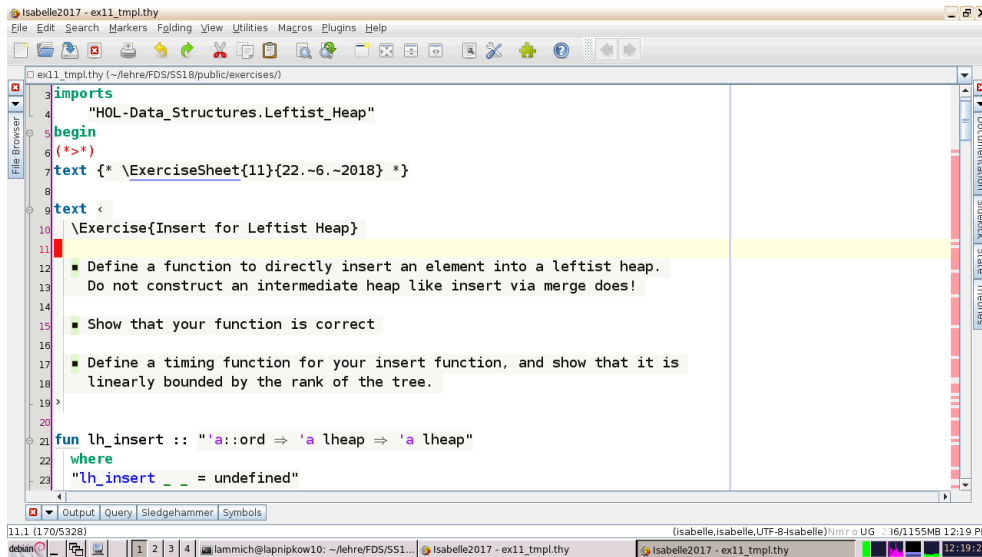
Date: Fri Jun 22 12:18:58 CEST 2018

Duration: 92:56 min

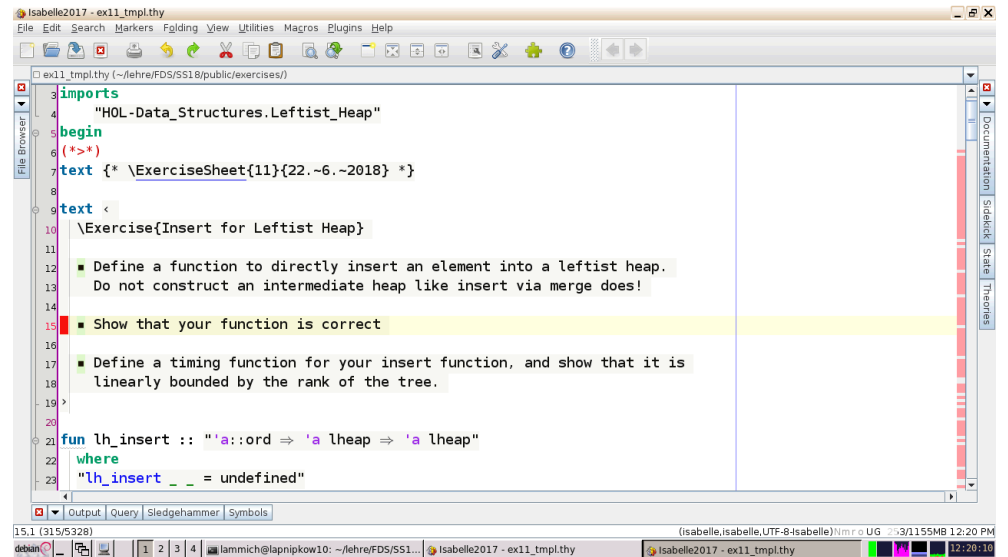
Pages: 106



```
1 imports
2   "HOL-Data_Structures.Leftist_Heap"
3 begin
4 (*>*)
5 text {* \ExerciseSheet{11}{22.-6.-2018} *}
6
7 text <
8   \Exercise{Insert for Leftist Heap}
9
10  ■ Define a function to directly insert an element into a leftist heap.
11  Do not construct an intermediate heap like insert via merge does!
12
13  ■ Show that your function is correct
14
15  ■ Define a timing function for your insert function, and show that it is
16  linearly bounded by the rank of the tree.
17
18 >
19
20 fun lh_insert :: "'a::ord ⇒ 'a lheap ⇒ 'a lheap"
21 where
22   "lh_insert _ _ = undefined"
```



```
1 imports
2   "HOL-Data_Structures.Leftist_Heap"
3 begin
4 (*>*)
5 text {* \ExerciseSheet{11}{22.-6.-2018} *}
6
7 text <
8   \Exercise{Insert for Leftist Heap}
9
10  ■ Define a function to directly insert an element into a leftist heap.
11  Do not construct an intermediate heap like insert via merge does!
12
13  ■ Show that your function is correct
14
15  ■ Define a timing function for your insert function, and show that it is
16  linearly bounded by the rank of the tree.
17
18 >
19
20 fun lh_insert :: "'a::ord ⇒ 'a lheap ⇒ 'a lheap"
21 where
22   "lh_insert _ _ = undefined"
```



```
1 imports
2   "HOL-Data_Structures.Leftist_Heap"
3 begin
4 (*>*)
5 text {* \ExerciseSheet{11}{22.-6.-2018} *}
6
7 text <
8   \Exercise{Insert for Leftist Heap}
9
10  ■ Define a function to directly insert an element into a leftist heap.
11  Do not construct an intermediate heap like insert via merge does!
12
13  ■ Show that your function is correct
14
15  ■ Define a timing function for your insert function, and show that it is
16  linearly bounded by the rank of the tree.
17
18 >
19
20 fun lh_insert :: "'a::ord ⇒ 'a lheap ⇒ 'a lheap"
21 where
22   "lh_insert _ _ = undefined"
```

```

Isabelle2017 - ex11_tmpl.thy
File Edit Search Markers Folding View Utilities Macros Plugins Help
ex11_tmpl.thy (~/lehre/FDS/SS18/public/exercises/)
15 Show that your function is correct
16
17
18 Define a timing function for your insert function, and show that it is
19 linearly bounded by the rank of the tree.
20
21 fun lh_insert :: "'a::ord ⇒ 'a lheap ⇒ 'a lheap"
22 where
23   "lh_insert _ = undefined"
24
25 lemma mset_lh_insert: "mset_tree (lh_insert x t) = mset_tree t + {# x #}"
26 sorry
27
28 lemma "heap t ⇒ heap (lh_insert x t)"
29 sorry
30
31 lemma "ltree t ⇒ ltree (lh_insert x t)"
32 sorry
33
34
35 fun t_lh_insert :: "'a::ord ⇒ 'a lheap ⇒ nat"

```

28.1 (650/5328) (isabelle.isabelle.UTF-8-isabelle)Nmr o UG 430/1155MB 12:21 PM

```

Isabelle2017 - Leftist_Heap.thy
File Edit Search Markers Folding View Utilities Macros Plugins Help
Leftist_Heap.thy ($ISABELLE_HOME/src/HOL/Data_Structures/)
23 fun rk :: "'a lheap ⇒ nat" where
24   "rk Leaf = 0" |
25   "rk (Node n _ _ ) = n"
26
27 text{* The invariants: *}
28
29 fun (in linorder) heap :: "('a,'b) tree ⇒ bool" where
30   "heap Leaf = True" |
31   "heap (Node _ l m r) =
32     (heap l ∧ heap r ∧ (∀x ∈ set_mset(mset_tree l + mset_tree r). m ≤ x))"
33
34 fun ltree :: "'a lheap ⇒ bool" where
35   "ltree Leaf = True" |
36   "ltree (Node n l a r) =
37     (n = rank r + 1 ∧ rank l ≥ rank r ∧ ltree l & ltree r)"
38
39 definition node :: "'a lheap ⇒ 'a ⇒ 'a lheap ⇒ 'a lheap" where
40   "node l a r =
41     (let rl = rk l; rr = rk r
42      in if rl ≥ rr then Node (rr+1) l a r else Node (rl+1) r a l)"
43

```

39.1 (823/7567) (isabelle.isabelle.UTF-8-isabelle)Nmr o UG 607/1155MB 12:21 PM

```

Isabelle2017 - Leftist_Heap.thy
File Edit Search Markers Folding View Utilities Macros Plugins Help
Leftist_Heap.thy ($ISABELLE_HOME/src/HOL/Data_Structures/)
27 text{* The invariants: *}
28
29 fun (in linorder) heap :: "('a,'b) tree ⇒ bool" where
30   "heap Leaf = True" |
31   "heap (Node _ l m r) =
32     (heap l ∧ heap r ∧ (∀x ∈ set_mset(mset_tree l + mset_tree r). m ≤ x))"
33
34 fun ltree :: "'a lheap ⇒ bool" where
35   "ltree Leaf = True" |
36   "ltree (Node n l a r) =
37     (n = rank r + 1 ∧ rank l ≥ rank r ∧ ltree l & ltree r)"
38
39 definition node :: "'a lheap ⇒ 'a ⇒ 'a lheap ⇒ 'a lheap" where
40   "node l a r =
41     (let rl = rk l; rr = rk r
42      in if rl ≥ rr then Node (rr+1) l a r else Node (rl+1) r a l)"
43
44 fun get_min :: "'a lheap ⇒ 'a" where
45   "get_min(Node n l a r) = a"
46
47 text <For function <merge>.>

```

39.1 (823/7567) (isabelle.isabelle.UTF-8-isabelle)Nmr o UG 607/1158MB 12:22 PM

```

Isabelle2017 - Leftist_Heap.thy
File Edit Search Markers Folding View Utilities Macros Plugins Help
Leftist_Heap.thy ($ISABELLE_HOME/src/HOL/Data_Structures/)
74
75 subsection "Lemmas"
76
77 lemma mset_tree_empty: "mset_tree t = {#} ↔ t = Leaf"
78 by(cases t) auto
79
80 lemma rk_eq_rank[simp]: "ltree t ⇒ rk t = rank t"
81 by(cases t) auto
82
83 lemma ltree_node: "ltree (node l a r) ↔ ltree l ∧ ltree r"
84 by(auto simp add: node_def)
85
86 lemma heap_node: "heap (node l a r) ↔
87   heap l ∧ heap r ∧ (∀x ∈ set_mset(mset_tree l + mset_tree r). a ≤ x)"
88 by(auto simp add: node_def)
89
90 subsection "Functional Correctness"
91
92 lemma mset_merge: "mset_tree (merge h1 h2) = mset_tree h1 + mset_tree h2"
93 by (induction h1 h2 rule: merge.induct) (auto simp add: node_def ac_simps)

```

86.1 (2166/7567) (isabelle.isabelle.UTF-8-isabelle)Nmr o UG 607/1158MB 12:23 PM

```
Isabelle2017 - ex11_tmpl.thy
File Edit Search Markers Folding View Utilities Macros Plugins Help
ex11_tmpl.thy (~/lehre/FDS/SS18/public/exercises/)
14
15
16
17
18
19
20
21 fun lh_insert :: "'a::ord => 'a lheap => 'a lheap"
22   where
23   "lh_insert _ _ = undefined"
24
25 lemma mset_lh_insert: "mset_tree (lh_insert x t) = mset_tree t + {# x #}"
26   sorry
27
28 lemma "heap t => heap (lh_insert x t)"
29   sorry
30
31 lemma "ltree t => ltree (lh_insert x t)"
32   sorry
33
34
Output Query Sledgehammer Symbols
23.8 (543/5328) (Isabelle.isabelle.UTF-8-Isabelle)Nmr o UG 436/1158MB 12:24 PM
debian 1 2 3 4 iamlich@lapnikow10: ~/lehre/FDS/SS1... Isabelle2017 - ex11_tmpl.thy Isabelle2017 - ex11_tmpl.thy 12:24:45
```

```
Isabelle2017 - ex11_tmpl.thy (modified)
File Edit Search Markers Folding View Utilities Macros Plugins Help
ex11_tmpl.thy (~/lehre/FDS/SS18/public/exercises/)
14
15
16
17
18
19
20
21 fun lh_insert :: "'a::ord => 'a lheap => 'a lheap"
22   where
23   "lh_insert x () = <"
24
25 lemma mset_lh_insert: "mset_tree (lh_insert x t) = mset_tree t + {# x #}"
26   sorry
27
28 lemma "heap t => heap (lh_insert x t)"
29   sorry
30
31 lemma "ltree t => ltree (lh_insert x t)"
32   sorry
33
34
Output Query Sledgehammer Symbols
23.22 (557/5321) (Isabelle.isabelle.UTF-8-Isabelle)Nmr o UG 688/1158MB 12:25 PM
debian 1 2 3 4 iamlich@lapnikow10: ~/lehre/FDS/SS1... Isabelle2017 - ex11_tmpl.thy (modified) Isabelle2017 - ex11_tmpl.thy (modified) 12:25:56
```

```
Isabelle2017 - ex11_tmpl.thy (modified)
File Edit Search Markers Folding View Utilities Macros Plugins Help
ex11_tmpl.thy (~/lehre/FDS/SS18/public/exercises/)
14
15
16
17
18
19
20
21 fun lh_insert :: "'a::ord => 'a lheap => 'a lheap"
22   where
23   "lh_insert x () = (1, (), x, ())"
24   "lh_insert x (_, l, a, r) = node"
25
26 lemma mset_lh_insert: "mset_tree (lh_insert x t) = mset_tree t + {# x #}"
27   sorry
28
29 lemma "heap t => heap (lh_insert x t)"
30   sorry
31
32 lemma "ltree t => ltree (lh_insert x t)"
33   sorry
34
Output Query Sledgehammer Symbols
24.32 (607/5371) (Isabelle.isabelle.UTF-8-Isabelle)Nmr o UG 92/1340MB 12:29 PM
debian 1 2 3 4 iamlich@lapnikow10: ~/lehre/FDS/SS1... Isabelle2017 - ex11_tmpl.thy (modified) Isabelle2017 - ex11_tmpl.thy (modified) 12:29:03
```

```
Isabelle2017 - ex11_tmpl.thy
File Edit Search Markers Folding View Utilities Macros Plugins Help
ex11_tmpl.thy (~/lehre/FDS/SS18/public/exercises/)
14
15
16
17
18
19
20
21 fun lh_insert :: "'a::ord => 'a lheap => 'a lheap"
22   where
23   "lh_insert x () = (1, (), x, ())"
24   "lh_insert x (_, l, a, r) = (if a < x then node l a (lh_insert x r) else node l x (lh_insert a r))"
25
26 lemma mset_lh_insert: "mset_tree (lh_insert x t) = mset_tree t + {# x #}"
27   sorry
28
29 lemma "heap t => heap (lh_insert x t)"
30   sorry
31
32 lemma "ltree t => ltree (lh_insert x t)"
33   sorry
34
Output Query Sledgehammer Symbols
27.1 (748/5435) (Isabelle.isabelle.UTF-8-Isabelle)Nmr o UG 92/1344MB 12:30 PM
debian 1 2 3 4 iamlich@lapnikow10: ~/lehre/FDS/SS1... Isabelle2017 - ex11_tmpl.thy Isabelle2017 - ex11_tmpl.thy 12:30:22
```

```

Isabelle2017 - ex11_tmpl.thy (modified)
File Edit Search Markers Folding View Utilities Macros Plugins Help
ex11_tmpl.thy (~/lehre/FDS/SS18/public/exercises/)
16
17
18 Define a timing function for your insert function, and show that it is
19 linearly bounded by the rank of the tree.
20
21 fun lh_insert :: "'a::ord => 'a lheap => 'a lheap"
22 where
23 "lh_insert x () = (1, (), x, ())"
24 | "lh_insert x (_,l,a,r) = (if a<x then node l a (lh_insert x r) else node l x (lh_insert a r))"
25
26 Lemma mset_lh_insert: "mset_tree (lh_insert x t) = mset_tree t + {# x #}"
27 sorry
28
29 Lemma "heap t => heap (lh_insert x t)"
30 sorry
31
32 Lemma "ltree t => ltree (lh_insert x t)"
33 sorry
34
35 fun t_lh_insert :: "'a::ord => 'a lheap => nat"

```

27.4 (751/5431) (isabelle.isabelle.UTF-8-Isabelle)Nm r o UG 620/1344MB 12:30 PM

```

Isabelle2017 - ex11_tmpl.thy (modified)
File Edit Search Markers Folding View Utilities Macros Plugins Help
ex11_tmpl.thy (~/lehre/FDS/SS18/public/exercises/)
19
20
21 fun lh_insert :: "'a::ord => 'a lheap => 'a lheap"
22 where
23 "lh_insert x () = (1, (), x, ())"
24 | "lh_insert x (_,l,a,r) = (if a<x then node l a (lh_insert x r) else node l x (lh_insert a r))"
25
26 Lemma mset_lh_insert: "mset_tree (lh_insert x t) = mset_tree t + {# x #}"
27 apply (induction t)
28
29 Lemma "heap t => heap (lh_insert x t)"
30 sorry
31
32 Lemma "ltree t => ltree (lh_insert x t)"
33 sorry
34
35
36 proof (prove)
37 goal (2 subgoals):
38 1. mset_tree (lh_insert x ()) = mset_tree () + {#x#}
39 2.  $\bigwedge x1\ t1\ x3\ t2.$ 

```

27.21 (768/5449) Input/output complete (isabelle.isabelle.UTF-8-Isabelle)Nm r o UG 112/1301MB 12:31 PM

```

Isabelle2017 - ex11_tmpl.thy (modified)
File Edit Search Markers Folding View Utilities Macros Plugins Help
ex11_tmpl.thy (~/lehre/FDS/SS18/public/exercises/)
19
20
21 fun lh_insert :: "'a::ord => 'a lheap => 'a lheap"
22 where
23 "lh_insert x () = (1, (), x, ())"
24 | "lh_insert x (_,l,a,r) = (if a<x then node l a (lh_insert x r) else node l x (lh_insert a r))"
25
26 Lemma mset_lh_insert: "mset_tree (lh_insert x t) = mset_tree t + {# x #}"
27 apply (induction t arb)
28
29 Lemma "heap t => heap (lh_insert x t)"
30 sorry
31
32 Lemma "ltree t => ltree (lh_insert x t)"
33 sorry
34
35
36 proof (prove)
37 goal (2 subgoals):
38 1. mset_tree (lh_insert x ()) = mset_tree () + {#x#}
39 2.  $\bigwedge x1\ t1\ x3\ t2.$ 

```

27.26 (773/5454) (isabelle.isabelle.UTF-8-Isabelle)Nm r o UG 118/1301MB 12:31 PM

```

Isabelle2017 - ex11_tmpl.thy (modified)
File Edit Search Markers Folding View Utilities Macros Plugins Help
ex11_tmpl.thy (~/lehre/FDS/SS18/public/exercises/)
19
20
21 fun lh_insert :: "'a::ord => 'a lheap => 'a lheap"
22 where
23 "lh_insert x () = (1, (), x, ())"
24 | "lh_insert x (_,l,a,r) = (if a<x then node l a (lh_insert x r) else node l x (lh_insert a r))"
25
26 Lemma mset_lh_insert: "mset_tree (lh_insert x t) = mset_tree t + {# x #}"
27 apply (induction t arbitrary: x)
28 apply (auto)
29
30
31 proof (prove)
32 goal (2 subgoals):
33 1.  $\bigwedge t1\ x3\ t2\ x.$ 
34 [ $\bigwedge x. mset\_tree (lh\_insert\ x\ t1) = add\_mset\ x\ (mset\_tree\ t1);$ 
35  $\bigwedge x. mset\_tree (lh\_insert\ x\ t2) = add\_mset\ x\ (mset\_tree\ t2); x3 < x]$ 
36  $\implies mset\_tree (node\ t1\ x3\ (lh\_insert\ x\ t2)) = add\_mset\ x\ (add\_mset\ x3\ (mset\_tree\ t1 + mset\_tree\ t2))$ 
37 2.  $\bigwedge t1\ x3\ t2\ x.$ 
38 [ $\bigwedge x. mset\_tree (lh\_insert\ x\ t1) = add\_mset\ x\ (mset\_tree\ t1).$ 

```

28.15 (797/5477) Input/output complete (isabelle.isabelle.UTF-8-Isabelle)Nm r o UG 118/1301MB 12:32 PM

```

Isabelle2017 - ex11_tmpl.thy (modified)
File Edit Search Markers Folding View Utilities Macros Plugins Help
ex11_tmpl.thy (~lehre/FDS/SS18/public/exercises/)
19 >
20
21 fun lh_insert :: "'a::ord ⇒ 'a lheap ⇒ 'a lheap"
22   where
23     "lh_insert x () = (1, ⟨, x, ⟨) )"
24     | "lh_insert x (⟦, l, a, r) = (if a < x then node l a (lh_insert x r) else node l x (lh_insert a r))"
25
26 Lemma mset_lh_insert: "mset_tree (lh_insert x t) = mset_tree t + {# x #}"
27   apply (induction t arbitrary: x)
28   apply (auto)
29
proof (prove)
goal (2 subgoals):
1.  $\wedge t1\ x3\ t2\ x.$ 
    $\llbracket \wedge x. \text{mset\_tree } (\text{lh\_insert } x\ t1) = \text{add\_mset } x\ (\text{mset\_tree } t1);$ 
    $\wedge x. \text{mset\_tree } (\text{lh\_insert } x\ t2) = \text{add\_mset } x\ (\text{mset\_tree } t2); x3 < x \rrbracket$ 
    $\implies \text{mset\_tree } (\text{node } t1\ x3\ (\text{lh\_insert } x\ t2)) = \text{add\_mset } x\ (\text{add\_mset } x3\ (\text{mset\_tree } t1 + \text{mset\_tree } t2))$ 
2.  $\wedge t1\ x3\ t2\ x.$ 
    $\llbracket \wedge x. \text{mset\_tree } (\text{lh\_insert } x\ t1) = \text{add\_mset } x\ (\text{mset\_tree } t1).$ 

```

```

Isabelle2017 - ex11_tmpl.thy (modified)
File Edit Search Markers Folding View Utilities Macros Plugins Help
ex11_tmpl.thy (~lehre/FDS/SS18/public/exercises/)
19 >
20
21 fun lh_insert :: "'a::ord ⇒ 'a lheap ⇒ 'a lheap"
22   where
23     "lh_insert x () = (1, ⟨, x, ⟨) )"
24     | "lh_insert x (⟦, l, a, r) = (if a < x then node l a (lh_insert x r) else node l x (lh_insert a r))"
25
26 Lemma mset_lh_insert: "mset_tree (lh_insert x t) = mset_tree t + {# x #}"
27   apply (induction t arbitrary: x)
28   apply (auto)
29
proof (prove)
goal (2 subgoals):
1.  $\wedge t1\ x3\ t2\ x.$ 
    $\llbracket \wedge x. \text{mset\_tree } (\text{lh\_insert } x\ t1) = \text{add\_mset } x\ (\text{mset\_tree } t1);$ 
    $\wedge x. \text{mset\_tree } (\text{lh\_insert } x\ t2) = \text{add\_mset } x\ (\text{mset\_tree } t2); x3 < x \rrbracket$ 
    $\implies \text{mset\_tree } (\text{node } t1\ x3\ (\text{lh\_insert } x\ t2)) = \text{add\_mset } x\ (\text{add\_mset } x3\ (\text{mset\_tree } t1 + \text{mset\_tree } t2))$ 
2.  $\wedge t1\ x3\ t2\ x.$ 
    $\llbracket \wedge x. \text{mset\_tree } (\text{lh\_insert } x\ t1) = \text{add\_mset } x\ (\text{mset\_tree } t1).$ 

```

```

Isabelle2017 - ex11_tmpl.thy (modified)
File Edit Search Markers Folding View Utilities Macros Plugins Help
ex11_tmpl.thy (~lehre/FDS/SS18/public/exercises/)
19 >
20
21 fun lh_insert :: "'a::ord ⇒ 'a lheap ⇒ 'a lheap"
22   where
23     "lh_insert x () = (1, ⟨, x, ⟨) )"
24     | "lh_insert x (⟦, l, a, r) = (if a < x then node l a (lh_insert x r) else node l x (lh_insert a r))"
25
26 Lemma mset_lh_insert: "mset_tree (lh_insert x t) = mset_tree t + {# x #}"
27   apply (induction t arbitrary: x)
28   apply (auto)
29
proof (prove)
goal (2 subgoals):
1.  $\wedge x. \text{mset\_tree } (\text{lh\_insert } x\ ()) = \text{mset\_tree } () + \{#x\#}$ 
2.  $\wedge x1\ t1\ x3\ t2\ x.$ 
    $\llbracket \wedge x. \text{mset\_tree } (\text{lh\_insert } x\ t1) = \text{mset\_tree } t1 + \{#x\#};$ 
    $\wedge x. \text{mset\_tree } (\text{lh\_insert } x\ t2) = \text{mset\_tree } t2 + \{#x\#} \rrbracket$ 
    $\implies \text{mset\_tree } (\text{lh\_insert } x\ (x1, t1, x3, t2)) = \text{mset\_tree } (x1, t1, x3, t2) + \{#x\#}$ 

```

```

Isabelle2017 - ex11_tmpl.thy (modified)
File Edit Search Markers Folding View Utilities Macros Plugins Help
ex11_tmpl.thy (~lehre/FDS/SS18/public/exercises/)
31 >
32
33 Lemma mset_lh_insert: "mset_tree (lh_insert x t) = mset_tree t + {# x #}"
34   by (induction t arbitrary: x) (auto)
35
36 Lemma "heap t ⇒ heap (lh_insert x t)"
37   find
38   find_theorems (keyword)
39   find_consts (keyword) (lh_insert x t)"
40   find_unused_assms (keyword)
41
proof (prove)
goal (1 subgoal):
1. heap t ⇒ heap (lh_insert x t)

```

Isabelle2017 - ex11_tmpl.thy (modified)

```

31
32
33
34 Lemma mset_lh_insert: "mset_tree (lh_insert x t) = mset_tree t + {# x #}"
35 by (induction t arbitrary: x) (auto)
36
37 Lemma "heap t  $\implies$  heap (lh_insert x t)"
38 find_theorems heap node
39
40 Lemma "ltree t  $\implies$  ltree (lh_insert x t)"
41 sorry

```

find_theorems
"heap"
"node"

found 1 theorem(s):
Leftist_Heap.heap_node:
heap (node ?l ?a ?r) = (heap ?l \wedge heap ?r \wedge Multiset.Ball (mset_tree ?l + mset_tree ?r) (op \leq ?a))

38.26 (974/5607) (Isabelle, Isabelle, UTF-8-Isabelle)NmroUG 480/1345MB 12:36 PM

Isabelle2017 - ex11_tmpl.thy (modified)

```

31
32
33
34 Lemma mset_lh_insert: "mset_tree (lh_insert x t) = mset_tree t + {# x #}"
35 by (induction t arbitrary: x) (auto)
36
37 Lemma "heap t  $\implies$  heap (lh_insert x t)"
38 find_theorems heap node
39
40 Lemma "ltree t  $\implies$  ltree (lh_insert x t)"
41 sorry

```

proof (prove)
goal (1 subgoal):
1. heap t \implies heap (lh_insert x t)

35.26 (894/5607) (Isabelle, Isabelle, UTF-8-Isabelle)NmroUG 551/1345MB 12:36 PM

Isabelle2017 - ex11_tmpl.thy (modified)

```

32
33
34 Lemma mset_lh_insert: "mset_tree (lh_insert x t) = mset_tree t + {# x #}"
35 by (induction t arbitrary: x) (auto)
36
37 Lemma "heap t  $\implies$  heap (lh_insert x t)"
38 apply
39
40 Lemma "ltree t  $\implies$  ltree (lh_insert x t)"
41 sorry

```

fun t_lh_insert :: "'a::ord \implies 'a heap \implies nat"

proof (prove)
goal (1 subgoal):
1. heap t \implies heap (lh_insert x t)
Outer syntax error: keyword "(" expected,
but end-of-input was found

35.9 (879/5590) (Isabelle, Isabelle, UTF-8-Isabelle)NmroUG 724/1345MB 12:37 PM

Isabelle2017 - ex11_tmpl.thy

```

32
33
34 Lemma mset_lh_insert: "mset_tree (lh_insert x t) = mset_tree t + {# x #}"
35 by (induction t arbitrary: x) (auto)
36
37 Lemma "heap t  $\implies$  heap (lh_insert x t)"
38 apply (induction t arbitrary: x)
39 apply (auto simp: heap_node)
40

```

proof (prove)
goal (3 subgoals):
1. $\wedge t1\ x3\ t2\ x\ xa.$
[[$\wedge x.$ heap (lh_insert x t1); $\wedge x.$ heap (lh_insert x t2); heap t1; heap t2;
 $\forall x \in \text{set_mset} (\text{mset_tree } t1) \cup \text{set_mset} (\text{mset_tree } t2). x3 \leq x; x3 < x;$
 $xa \in \# \text{mset_tree } (\text{lh_insert } x\ t2)$]]
 $\implies x3 \leq xa.$
2. $\wedge t1\ x3\ t2\ x\ xa.$
[[$\wedge x.$ heap (lh_insert x t1); $\wedge x.$ heap (lh_insert x t2); heap t1; heap t2;

39.30 (1013/5647) (Isabelle, Isabelle, UTF-8-Isabelle)NmroUG 997/1344MB 12:38 PM

```

Isabelle2017 - ex11_tmpl.thy
File Edit Search Markers Folding View Utilities Magros Plugins Help
ex11_tmpl.thy (~/lehre/FDS/SS18/public/exercises/)
32
33
34 Lemma mset_lh_insert: "mset_tree (lh_insert x t) = mset_tree t + {# x #}"
35 by (induction t arbitrary: x) (auto)
36
37 Lemma "heap t  $\implies$  heap (lh_insert x t)"
38 apply (induction t arbitrary: x)
39 apply (auto simp: heap_node)
40
proof (prove)
goal (3 subgoals):
1.  $\bigwedge t1\ x3\ t2\ x\ xa.$ 
 $[\bigwedge x. \text{heap } (lh\_insert\ x\ t1); \bigwedge x. \text{heap } (lh\_insert\ x\ t2); \text{heap } t1; \text{heap } t2;$ 
 $\forall x \in \text{set\_mset } (mset\_tree\ t1) \cup \text{set\_mset } (mset\_tree\ t2). x3 \leq x; x3 < x;$ 
 $xa \in \# mset\_tree\ (lh\_insert\ x\ t2)]$ 
 $\implies x3 \leq xa$ 
2.  $\bigwedge t1\ x3\ t2\ x\ xa.$ 
 $[\bigwedge x. \text{heap } (lh\_insert\ x\ t1); \bigwedge x. \text{heap } (lh\_insert\ x\ t2); \text{heap } t1; \text{heap } t2;$ 

```

```

Isabelle2017 - ex11_tmpl.thy
File Edit Search Markers Folding View Utilities Magros Plugins Help
ex11_tmpl.thy (~/lehre/FDS/SS18/public/exercises/)
32
33
34 Lemma mset_lh_insert: "mset_tree (lh_insert x t) = mset_tree t + {# x #}"
35 by (induction t arbitrary: x) (auto)
36
37 Lemma "heap t  $\implies$  heap (lh_insert x t)"
38 apply (induction t arbitrary: x)
39 apply (auto simp: heap_node)
40
proof (prove)
goal (1 subgoal):
1. mset_tree (lh_insert x t) = mset_tree t + {#x#}

```

```

Isabelle2017 - ex11_tmpl.thy
File Edit Search Markers Folding View Utilities Magros Plugins Help
ex11_tmpl.thy (~/lehre/FDS/SS18/public/exercises/)
34 Lemma mset_lh_insert: "mset_tree (lh_insert x t) = mset_tree t + {# x #}"
35 by (induction t arbitrary: x) (auto)
36
37 Lemma "heap t  $\implies$  heap (lh_insert x t)"
38 apply (induction t arbitrary: x)
39 apply (auto simp: heap_node mset_lh_insert)
40
41 Lemma "ltree t  $\implies$  ltree (lh_insert x t)"
42 sorry
proof (prove)
goal (2 subgoals):
1.  $\bigwedge t1\ x3\ t2\ x\ xa.$ 
 $[\bigwedge x. \text{heap } (lh\_insert\ x\ t1); \bigwedge x. \text{heap } (lh\_insert\ x\ t2); \text{heap } t1; \text{heap } t2;$ 
 $\forall x \in \text{set\_mset } (mset\_tree\ t1) \cup \text{set\_mset } (mset\_tree\ t2). x3 \leq x; \neg x3 < x; xa \in \# mset\_tree\ t1]$ 
 $\implies x \leq xa$ 
2.  $\bigwedge t1\ x3\ t2\ x\ xa.$ 
 $[\bigwedge x. \text{heap } (lh\_insert\ x\ t1); \bigwedge x. \text{heap } (lh\_insert\ x\ t2); \text{heap } t1; \text{heap } t2;$ 
 $\forall x \in \text{set\_mset } (mset\_tree\ t1) \cup \text{set\_mset } (mset\_tree\ t2). x3 \leq x; \neg x3 < x; xa \in \# mset\_tree\ t2]$ 

```

```

Isabelle2017 - ex11_tmpl.thy
File Edit Search Markers Folding View Utilities Magros Plugins Help
ex11_tmpl.thy (~/lehre/FDS/SS18/public/exercises/)
36 Lemma "heap t  $\implies$  heap (lh_insert x t)"
37 apply (induction t arbitrary: x)
38 apply (force simp: heap_node mset_lh_insert)
39 done
40
41
42
43 Lemma "ltree t  $\implies$  ltree (lh_insert x t)"
44 sorry
theorem heap ?t  $\implies$  heap (lh_insert ?x ?t)

```

```

Isabelle2017 - ex11_tmpl.thy
File Edit Search Markers Folding View Utilities Macros Plugins Help
ex11_tmpl.thy (~/lehre/FDS/SS18/public/exercises/)
43 lemma "ltree t ==> ltree (lh_insert x t)"
44   apply (induction t arbitrary: x)
45   apply (auto simp: ltree_node)
46   done
47
48 fun t_lh_insert :: "'a::ord => 'a lheap => nat"
49   where
50   "t_lh_insert _ _ = undefined"
51
theorem ltree ?t ==> ltree (lh_insert ?x ?t)

```

47.1 (1155/5737) (isabelle,isabelle,UTF-8-Isabelle)NmroUG 363/1347MB 12:43 PM

```

Isabelle2017 - ex11_tmpl.thy
File Edit Search Markers Folding View Utilities Macros Plugins Help
ex11_tmpl.thy (~/lehre/FDS/SS18/public/exercises/)
30
31
32
33
34 lemma mset_lh_insert: "mset_tree (lh_insert x t) = mset_tree t + {# x #}"
35   by (induction t arbitrary: x) (auto)
36
37 lemma "heap t ==> heap (lh_insert x t)"
38   apply (induction t arbitrary: x)

```

35.1 (871/5737) (isabelle,isabelle,UTF-8-Isabelle)NmroUG 748/1447MB 12:44 PM

```

Isabelle2017 - ex11_tmpl.thy
File Edit Search Markers Folding View Utilities Macros Plugins Help
ex11_tmpl.thy (~/lehre/FDS/SS18/public/exercises/)
45   apply (auto simp: ltree_node)
46   done
47
48 fun t_lh_insert :: "'a::ord => 'a lheap => nat"
49   where
50   "t_lh_insert _ _ = undefined"
51
52 lemma "t_lh_insert x t <= rank t + 1"
53   sorry
54
55 text <
56   \Exercise{Bootstrapping a Priority Queue}
57
58   Given a generic priority queue implementation with
59   <0(1)> <empty>, <is_empty> operations, <0(f_1 n)> insert,

```

50.1 (1210/5737) (isabelle,isabelle,UTF-8-Isabelle)NmroUG 8/1345MB 12:44 PM

```

Isabelle2017 - ex11_tmpl.thy
File Edit Search Markers Folding View Utilities Macros Plugins Help
ex11_tmpl.thy (~/lehre/FDS/SS18/public/exercises/)
45   apply (auto simp: ltree_node)
46   done
47
48 (*
49   "lh_insert x () = (1, (), x, ())"
50   | "lh_insert x (_,l,a,r) = (if a<x then node l a (lh_insert x r) else node l x (lh_insert a r))"
51   *)
52
53 fun t_lh_insert :: "'a::ord => 'a lheap => nat"
54   where
55   "t_lh_insert _ _ = undefined"
56
57 lemma "t_lh_insert x t <= rank t + 1"
58   sorry
59

```

49.40 (1198/5881) (isabelle,isabelle,UTF-8-Isabelle)NmroUG 720/1445MB 12:45 PM


```

Isabelle2017 - ex11_tmpl.thy
File Edit Search Markers Folding View Utilities Macros Plugins Help
ex11_tmpl.thy (~/lehre/FDS/SS18/public/exercises/)
45 apply (auto simp: ltree_node)
46 done
47
48 (*
49 "lh_insert x () = (1, (), x, ())"
50 | "lh_insert x (_,l,a,r) = (if a<x then node l a (lh_insert x r) else node l x (lh_insert a r))"
51 *)
52
53 fun t_lh_insert :: "'a::ord ⇒ 'a lheap ⇒ nat"
54 where
55 "t_lh_insert _ = undefined"
56
57 Lemma "t_lh_insert x t ≤ rank t + 1"
58 sorry
59
consts
t_lh_insert :: "'a ⇒ ('a, nat) tree ⇒ nat"
Found termination order: "{}"

```

50.1 (1199/5981) (Isabelle.isabelle.UTF-8-Isabelle)Nmr o UG 81.4/1350MB 12:46 PM
debian 1 2 3 4 iamnich@lapnikow10: ~/lehre/FDS/SS1... Isabelle2017 - ex11_tmpl.thy Isabelle2017 - ex11_tmpl.thy 12:46:33

```

Isabelle2017 - ex11_tmpl.thy
File Edit Search Markers Folding View Utilities Macros Plugins Help
ex11_tmpl.thy (~/lehre/FDS/SS18/public/exercises/)
47 (*
48 "lh_insert x () = (1, (), x, ())"
49 | "lh_insert x (_,l,a,r) = (if a<x then node l a (lh_insert x r) else node l x (lh_insert a r))"
50 *)
51
52 fun t_lh_insert :: "'a::ord ⇒ 'a lheap ⇒ nat"
53 where
54 "t_lh_insert x () = 1"
55 | "t_lh_insert x (_,l,a,r) = (if a<x then 1 + (t_lh_insert x r) else 1 + (t_lh_insert a r))"
56
57 Lemma "t_lh_insert x t ≤ rank t + 1"
58 sorry
59
text <

```

57.1 (1472/5967) (Isabelle.isabelle.UTF-8-Isabelle)Nmr o UG 40.8/1350MB 12:48 PM
debian 1 2 3 4 iamnich@lapnikow10: ~/lehre/FDS/SS1... Isabelle2017 - ex11_tmpl.thy Isabelle2017 - ex11_tmpl.thy 12:48:00

```

Isabelle2017 - ex11_tmpl.thy
File Edit Search Markers Folding View Utilities Macros Plugins Help
ex11_tmpl.thy (~/lehre/FDS/SS18/public/exercises/)
47 (*
48 "lh_insert x () = (1, (), x, ())"
49 | "lh_insert x (_,l,a,r) = (if a<x then node l a (lh_insert x r) else node l x (lh_insert a r))"
50 *)
51
52 fun t_lh_insert :: "'a::ord ⇒ 'a lheap ⇒ nat"
53 where
54 "t_lh_insert x () = 1"
55 | "t_lh_insert x (_,l,a,r) = (if a<x then 1 + (t_lh_insert x r) else 1 + (t_lh_insert a r))"
56
57 Lemma "t_lh_insert x t ≤ rank t + 1"
58 sorry
59
text <

```

58.26 (1498/5967) (Isabelle.isabelle.UTF-8-Isabelle)Nmr o UG 42.2/1350MB 12:48 PM
debian 1 2 3 4 iamnich@lapnikow10: ~/lehre/FDS/SS1... Isabelle2017 - ex11_tmpl.thy Isabelle2017 - ex11_tmpl.thy 12:48:18

proof (prove)
goal (1 subgoal):
1. t_lh_insert x t ≤ rank t + 1

```

Isabelle2017 - ex11_tmpl.thy
File Edit Search Markers Folding View Utilities Macros Plugins Help
ex11_tmpl.thy (~/lehre/FDS/SS18/public/exercises/)
47 (*
48 "lh_insert x () = (1, (), x, ())"
49 | "lh_insert x (_,l,a,r) = (if a<x then node l a (lh_insert x r) else node l x (lh_insert a r))"
50 *)
51
52 fun t_lh_insert :: "'a::ord ⇒ 'a lheap ⇒ nat"
53 where
54 "t_lh_insert x () = 1"
55 | "t_lh_insert x (_,l,a,r) = (if a<x then 1 + (t_lh_insert x r) else 1 + (t_lh_insert a r))"
56
57 Lemma "t_lh_insert x t ≤ rank t + 1"
58 sorry
59
text <

```

56.36 (1414/5967) (Isabelle.isabelle.UTF-8-Isabelle)Nmr o UG 41.7/1350MB 12:48 PM
debian 1 2 3 4 iamnich@lapnikow10: ~/lehre/FDS/SS1... Isabelle2017 - ex11_tmpl.thy Isabelle2017 - ex11_tmpl.thy 12:48:00

consts
t_lh_insert :: "'a ⇒ ('a, nat) tree ⇒ nat"
Found termination order: "(λp. size (snd p)) < *mlex* {}"

```

 Isabelle2017 - ex11_tmpl.thy
 File Edit Search Markers Folding View Utilities Macros Plugins Help
 ex11_tmpl.thy (~/lehre/FDS/SS18/public/exercises/)
 48 (*
 49   "lh_insert x () = (1, (), x, ())"
 50 | "lh_insert x (_,l,a,r) = (if a<x then node l a (lh_insert x r) else node l x (lh_insert a r))"
 51 *)
 52
 53 fun t_lh_insert :: "'a::ord ⇒ 'a heap ⇒ nat"
 54 where
 55   "t_lh_insert x () = 1"
 56 | "t_lh_insert x (_,l,a,r) = (if a<x then 1 + (t_lh_insert x r) else 1 + (t_lh_insert a r))"
 57
 58 lemma "t_lh_insert x t ≤ rank t + 1"
 59 sorry
 60
 61 text <
 62 \Exercise{Bootstrapping a Priority Queue}
 63
 64 proof (prove)
 65 goal (1 subgoal):
 66 1. t_lh_insert x t ≤ rank t + 1
 67
 68 Output Query Sledgehammer Symbols
 59.1 (1510/5967) (isabelle.isabelle.UTF-8-isabelle)Nmr o UG 461/1350MB 12:49 PM
  
```

```

 Isabelle2017 - ex11_tmpl.thy
 File Edit Search Markers Folding View Utilities Macros Plugins Help
 ex11_tmpl.thy (~/lehre/FDS/SS18/public/exercises/)
 51 *)
 52
 53 fun t_lh_insert :: "'a::ord ⇒ 'a heap ⇒ nat"
 54 where
 55   "t_lh_insert x () = 1"
 56 | "t_lh_insert x (_,l,a,r) = (if a<x then 1 + (t_lh_insert x r) else 1 + (t_lh_insert a r))"
 57
 58 lemma "t_lh_insert x t ≤ rank t + 1"
 59
 60 text <
 61 \Exercise{Bootstrapping a Priority Queue}
 62
 63 Given a generic priority queue implementation with
 64 <O(1)> <empty>, <is_empty> operations, <O(f1 n)> insert,
 65
 66 proof (prove)
 67 goal (1 subgoal):
 68 1. t_lh_insert x t ≤ rank t + 1
 69
 70 Output Query Sledgehammer Symbols
 56.1 (1379/5960) (isabelle.isabelle.UTF-8-isabelle)Nmr o UG 817/1350MB 12:50 PM
  
```

```

 Isabelle2017 - ex11_tmpl.thy
 File Edit Search Markers Folding View Utilities Macros Plugins Help
 ex11_tmpl.thy (~/lehre/FDS/SS18/public/exercises/)
 52
 53 fun t_lh_insert :: "'a::ord ⇒ 'a heap ⇒ nat"
 54 where
 55   "t_lh_insert x () = 1"
 56 | "t_lh_insert x (_,l,a,r) = (if a<x then 1 + (t_lh_insert x r) else 1 + (t_lh_insert a r))"
 57
 58 lemma "t_lh_insert x t ≤ rank t + 1"
 59 apply (induction t arbitrary: x)
 60 apply auto
 61 done
 62
 63 text <
 64 \Exercise{Bootstrapping a Priority Queue}
 65
 66 consts
 67 t_lh_insert :: "'a ⇒ ('a, nat) tree ⇒ nat"
 68 Found termination order: "(λp. size (snd p)) <*>mlex*> {}"
 69
 70 Output Query Sledgehammer Symbols
 55.1 (1354/6015) (isabelle.isabelle.UTF-8-isabelle)Nmr o UG 117/1349MB 12:51 PM
  
```

```

 Isabelle2017 - ex11_tmpl.thy
 File Edit Search Markers Folding View Utilities Macros Plugins Help
 ex11_tmpl.thy (~/lehre/FDS/SS18/public/exercises/)
 56 | "t_lh_insert x (_,l,a,r) = (if a<x then 1 + (t_lh_insert x r) else 1 + (t_lh_insert a r))"
 57
 58 lemma "t_lh_insert x t ≤ rank t + 1"
 59 apply (induction t arbitrary: x)
 60 apply auto
 61 done
 62
 63 text <
 64 \Exercise{Bootstrapping a Priority Queue}
 65
 66 Given a generic priority queue implementation with
 67 <O(1)> <empty>, <is_empty> operations, <O(f1 n)> insert,
 68 and <O(f2 n)> <get_min> and <del_min> operations.
 69
 70
 71 consts
 72 t_lh_insert :: "'a ⇒ ('a, nat) tree ⇒ nat"
 73 Found termination order: "(λp. size (snd p)) <*>mlex*> {}"
 74
 75 Output Query Sledgehammer Symbols
 67.1 (1619/6015) (isabelle.isabelle.UTF-8-isabelle)Nmr o UG 117/1349MB 12:51 PM
  
```

Isabelle2017 - ex11_tmpl.thy

```
62
63
64 text <
65 \Exercise{Bootstrapping a Priority Queue}
66
67 Given a generic priority queue implementation with
68 <0(1)> <empty>, <is_empty> operations, <0(f1 n)> insert,
69 and <0(f2 n)> <get_min> and <del_min> operations.
70
71 Derive an implementation with <0(1)> <get_min>, and the
72 asymptotic complexities of the other operations unchanged!
73
74 Hint: Store the current minimal element! As you know nothing
75 about <f1> and <f2>, you must not use <get_min/del_min>
76 in your new <insert> operation, and vice versa!
```

69.1 (1732/6015) (Isabelle, Isabelle, UTF-8-Isabelle) Nmr o UG 387/1349MB 12:52 PM

Isabelle2017 - ex11_tmpl.thy

```
62
63
64 text <
65 \Exercise{Bootstrapping a Priority Queue}
66
67 Given a generic priority queue implementation with
68 <0(1)> <empty>, <is_empty> operations, <0(f1 n)> insert,
69 and <0(f2 n)> <get_min> and <del_min> operations.
70
71 Derive an implementation with <0(1)> <get_min>, and the
72 asymptotic complexities of the other operations unchanged!
73
74 Hint: Store the current minimal element! As you know nothing
75 about <f1> and <f2>, you must not use <get_min/del_min>
76 in your new <insert> operation, and vice versa!
```

69.18 (1749/6015) (Isabelle, Isabelle, UTF-8-Isabelle) Nmr o UG 414/1349MB 12:52 PM

Isabelle2017 - ex11_tmpl.thy

```
66
67 Given a generic priority queue implementation with
68 <0(1)> <empty>, <is_empty> operations, <0(f1 n)> insert,
69 and <0(f2 n)> <get_min> and <del_min> operations.
70
71 Derive an implementation with <0(1)> <get_min>, and the
72 asymptotic complexities of the other operations unchanged!
73
74 Hint: Store the current minimal element! As you know nothing
75 about <f1> and <f2>, you must not use <get_min/del_min>
76 in your new <insert> operation, and vice versa!
77
78 text <For technical reasons, you have to define the new implementations type
79 outside the locale!>
80 datatype ('a, 's) bs_pq = PUT_CONSTRUCTORS_HERE
```

77.1 (2083/6015) (Isabelle, Isabelle, UTF-8-Isabelle) Nmr o UG 440/1349MB 12:53 PM

Isabelle2017 - ex11_tmpl.thy

```
67
68 Given a generic priority queue implementation with
69 <0(1)> <empty>, <is_empty> operations, <0(f1 n)> insert,
70 and <0(f2 n)> <get_min> and <del_min> operations.
71
72 Derive an implementation with <0(1)> <get_min>, and the
73 asymptotic complexities of the other operations unchanged!
74
75 Hint: Store the current minimal element! As you know nothing
76 about <f1> and <f2>, you must not use <get_min/del_min>
77 in your new <insert> operation, and vice versa!
78
79 text <For technical reasons, you have to define the new implementations type
80 outside the locale!>
81 datatype ('a, 's) bs_pq = PUT_CONSTRUCTORS_HERE
```

78.1 (2085/6015) (Isabelle, Isabelle, UTF-8-Isabelle) Nmr o UG 441/1349MB 12:54 PM

Isabelle2017 - ex11_tmpl.thy

```

72 asymptotic complexities of the other operations unchanged!
73
74 Hint: Store the current minimal element! As you know nothing
75 about <f1> and <f2>, you must not use <get_min/del_min>
76 in your new <insert> operation, and vice versa!
77
78
79 text <For technical reasons, you have to define the new implementations type
80 outside the locale!>
81 datatype ('a, 's) bs_pq = PUT_CONSTRUCTORS_HERE
82
83 locale Bs_Priority_Queue =
84   orig: Priority_Queue where
85   empty = orig_empty and
86   is_empty = orig_is_empty and

```

locale

Bs_Priority_Queue

83.1 (2238/6014) (isabelle.isabelle.UTF-8-isabelle)Nmr o UG 599/1349MB 12:56 PM

Isabelle2017 - ex11_tmpl.thy

```

83
84
85 locale Bs_Priority_Queue =
86   orig: Priority_Queue where
87   empty = orig_empty and
88   is_empty = orig_is_empty and
89   insert = orig_insert and
90   get_min = orig_get_min and
91   del_min = orig_del_min and
92   invar = orig_invar and
93   mset = orig_mset
94   for orig_empty orig_is_empty orig_insert orig_get_min orig_del_min orig_invar
95   and orig_mset :: "'s => 'a::linorder multiset"
96 begin
97   text <In here, the original implementation is available with the prefix <orig>, e.g. >

```

locale

Bs_Priority_Queue

86.1 (2264/6014) (isabelle.isabelle.UTF-8-isabelle)Nmr o UG 389/1353MB 12:57 PM

Isabelle2017 - ex11_tmpl.thy

```

84
85 locale Bs_Priority_Queue =
86   orig: Priority_Queue where
87   empty = orig_empty and
88   is_empty = orig_is_empty and
89   insert = orig_insert and
90   get_min = orig_get_min and
91   del_min = orig_del_min and
92   invar = orig_invar and
93   mset = orig_mset
94   for orig_empty orig_is_empty orig_insert orig_get_min orig_del_min orig_invar
95   and orig_mset :: "'s => 'a::linorder multiset"
96 begin
97   text <In here, the original implementation is available with the prefix <orig>, e.g. >
98   term orig_empty term orig_invar

```

locale

Bs_Priority_Queue

95.1 (2572/6014) (isabelle.isabelle.UTF-8-isabelle)Nmr o UG 760/1353MB 12:57 PM

Isabelle2017 - ex11_tmpl.thy

```

86
87 orig: Priority_Queue where
88   empty = orig_empty and
89   is_empty = orig_is_empty and
90   insert = orig_insert and
91   get_min = orig_get_min and
92   del_min = orig_del_min and
93   invar = orig_invar and
94   mset = orig_mset
95   for orig_empty orig_is_empty orig_insert orig_get_min orig_del_min orig_invar
96   and orig_mset :: "'s => 'a::linorder multiset"
97 begin
98   text <In here, the original implementation is available with the prefix <orig>, e.g. >
99   term orig_empty term orig_invar
100   thm orig_invar_empty

```

locale

Bs_Priority_Queue

96.6 (2625/6014) (isabelle.isabelle.UTF-8-isabelle)Nmr o UG 760/1353MB 12:57 PM

```

Isabelle2017 - ex11_tmpl.thy
File Edit Search Markers Folding View Utilities Macros Plugins Help
ex11_tmpl.thy (~/lehre/FDS/SS18/public/exercises/)
89 insert = orig_insert and
90 get_min = orig_get_min and
91 del_min = orig_del_min and
92 invar = orig_invar and
93 mset = orig_mset
94 for orig_empty orig_is_empty orig_insert orig_get_min orig_del_min orig_invar
95 and orig_mset :: "'s => 'a::linorder multiset"
96 begin
97 text <In here, the original implementation is available with the prefix <orig>, e.g. >
98 term orig_empty term orig_invar
99 thm orig.invar_empty
100
101 definition empty :: "('a,'s) bs_pq"
102 where "empty = undefined"
103
104 fun is_empty :: "('a,'s) bs_pq => bool"

```

97.23 (2648/6014) (isabelle.isabelle.UTF-8-isabelle)tmr o UG 40/1352MB 12:58 PM

```

Isabelle2017 - ex11_tmpl.thy (modified)
File Edit Search Markers Folding View Utilities Macros Plugins Help
ex11_tmpl.thy (~/lehre/FDS/SS18/public/exercises/)
82
83
84
85 locale Bs_Priority_Queue =
86 orig: Priority_Queue where
87 empty = orig_empty and
88 is_empty = orig_is_empty and
89 insert = orig_insert and
90 get_min = orig_get_min and
91 del_min = orig_del_min and
92 invar = orig_invar and
93 mset = orig_mset
94 for orig_empty orig_is_empty orig_insert orig_get_min orig_del_min orig_invar
95 and orig_mset :: "'q => 'a::linorder multiset"
96 begin
97 text <In here, the original implementation is available with the prefix <orig>, e.g. >

```

93.1 (2471/6018) (isabelle.isabelle.UTF-8-isabelle)tmr o UG 66/1351MB 1:00 PM

```

Isabelle2017 - ex11_tmpl.thy
File Edit Search Markers Folding View Utilities Macros Plugins Help
ex11_tmpl.thy (~/lehre/FDS/SS18/public/exercises/)
96 begin
97 text <In here, the original implementation is available with the prefix <orig>, e.g. >
98 term orig_empty term orig_invar
99 thm orig.invar_empty
100
101 fun mset :: "('a,'q) bs_pq => 'a multiset"
102 where
103 "mset Empty = {#}"
104 | "mset (Heap _ q) = orig_mset q"
105
106
107 fun invar :: "('a,'q) bs_pq => bool"
108 where
109 "invar _ = undefined"
110
111

```

104.20 (2871/6055) (isabelle.isabelle.UTF-8-isabelle)tmr o UG 45/1353MB 1:01 PM

```

consts
mset :: "('a,'q) bs_pq => 'a multiset"
Found termination order: "{}"

```

```

Isabelle2017 - ex11_tmpl.thy (modified)
File Edit Search Markers Folding View Utilities Macros Plugins Help
ex11_tmpl.thy (~/lehre/FDS/SS18/public/exercises/)
96 begin
97 text <In here, the original implementation is available with the prefix <orig>, e.g. >
98 term orig_empty term orig_invar
99 thm orig.invar_empty
100
101 fun mset :: "('a,'q) bs_pq => 'a multiset"
102 where
103 "mset Empty = {#}"
104 | "mset (Heap a q) = {#a#} + orig_mset q"
105
106
107 fun invar :: "('a,'q) bs_pq => bool"
108 where
109 "invar _ = undefined"
110
111

```

104.33 (2884/6053) (isabelle.isabelle.UTF-8-isabelle)tmr o UG 45/1353MB 1:02 PM

```

consts
mset :: "('a,'q) bs_pq => 'a multiset"
Found termination order: "{}"

```

```

Isabelle2017 - ex11_tmpl.thy (modified)
File Edit Search Markers Folding View Utilities Macros Plugins Help
ex11_tmpl.thy (~/lehre/FDS/SS18/public/exercises/)
begin
text <In here, the original implementation is available with the prefix <orig>, e.g. >
term orig_empty term orig_invar
thm orig_invar_empty

fun mset :: "('a,'q) bs_pq => 'a multiset"
  where
    "mset Empty = {#}"
    | "mset (Heap a q) = orig_mset q"

fun invar :: "('a,'q) bs_pq => bool"
  where
    "invar _ = undefined"

consts
mset :: "('a,'q) bs_pq => 'a multiset"
Found termination order: "{}"

```

104.26 (2877/6057) (isabelle.isabelle.UTF-8-isabelle)tmr o UG 61.3/1.353MB 1:02 PM

```

Isabelle2017 - ex11_tmpl.thy
File Edit Search Markers Folding View Utilities Macros Plugins Help
ex11_tmpl.thy (~/lehre/FDS/SS18/public/exercises/)
thm orig_invar_empty

fun mset :: "('a,'q) bs_pq => 'a multiset"
  where
    "mset Empty = {#}"
    | "mset (Heap _ q) = orig_mset q"

fun invar :: "('a,'q) bs_pq => bool"
  where
    "invar _ = undefined"

definition empty :: "('a,'q) bs_pq"
  where "empty = undefined"

consts
invar :: "('a,'q) bs_pq => bool"
Found termination order: "{}"

```

109.25 (2964/6055) (isabelle.isabelle.UTF-8-isabelle)tmr o UG 69.1/1.353MB 1:03 PM

```

Isabelle2017 - ex11_tmpl.thy
File Edit Search Markers Folding View Utilities Macros Plugins Help
ex11_tmpl.thy (~/lehre/FDS/SS18/public/exercises/)
thm orig_invar_empty

fun mset :: "('a,'q) bs_pq => 'a multiset"
  where
    "mset Empty = {#}"
    | "mset (Heap _ q) = orig_mset q"

fun invar :: "('a,'q) bs_pq => bool"
  where
    "invar _ = undefined"

definition empty :: "('a,'q) bs_pq"
  where "empty = undefined"

consts
invar :: "('a,'q) bs_pq => bool"
Found termination order: "{}"

```

109.14 (2963/6055) (isabelle.isabelle.UTF-8-isabelle)tmr o UG 69.1/1.353MB 1:03 PM

```

Isabelle2017 - ex11_tmpl.thy (modified)
File Edit Search Markers Folding View Utilities Macros Plugins Help
ex11_tmpl.thy (~/lehre/FDS/SS18/public/exercises/)
thm orig_invar_empty

fun mset :: "('a,'q) bs_pq => 'a multiset"
  where
    "mset Empty = {#}"
    | "mset (Heap _ q) = orig_mset q"

fun invar :: "('a,'q) bs_pq => bool"
  where
    "invar Empty <==> True"
    | "invar (Heap a q) = orig_invar q"

definition empty :: "('a,'q) bs_pq"
  where "empty = undefined"

Outer syntax error: keyword "|" expected,
but quoted string "invar (Heap a q) = orig_invar q" was found

```

109.24 (2963/6092) (isabelle.isabelle.UTF-8-isabelle)tmr o UG 70.4/1.305MB 1:03 PM

```

Isabelle2017 - ex11_tmpl.thy
File Edit Search Markers Folding View Utilities Macros Plugins Help
ex11_tmpl.thy (~/lehre/FDS/SS18/public/exercises/)
101 fun mset :: "('a,'q) bs_pq => 'a multiset"
102   where
103     "mset Empty = {#}"
104     | "mset (Heap _ q) = orig_mset q"
105
106
107 fun invar :: "('a,'q) bs_pq => bool"
108   where
109     "invar Empty <=> True"
110     | "invar (Heap a q) <=> orig_invar q & a ∈ #orig_mset q"
111
112
113 definition empty :: "('a,'q) bs_pq"
114   where "empty = undefined"
115
116 fun is_empty :: "('a,'q) bs_pq => bool"

consts
invar :: "('a,'q) bs_pq => bool"
Found termination order: "{}"

Output Query Sledgehammer Symbols
110.54 (3018/6109) (isabelle,isabelle.UTF-8-isabelle)tmr o UG 756/1305MB 1:04 PM

```

```

Isabelle2017 - ex11_tmpl.thy
File Edit Search Markers Folding View Utilities Macros Plugins Help
ex11_tmpl.thy (~/lehre/FDS/SS18/public/exercises/)
101 fun mset :: "('a,'q) bs_pq => 'a multiset"
102   where
103     "mset Empty = {#}"
104     | "mset (Heap _ q) = orig_mset q"
105
106
107 fun invar :: "('a,'q) bs_pq => bool"
108   where
109     "invar Empty <=> True"
110     | "invar (Heap a q) <=> orig_invar q & a ∈ #orig_mset q & a = Min_mset (orig_mset q)"
111
112
113 definition empty :: "('a,'q) bs_pq"
114   where "empty = undefined"
115
116 fun is_empty :: "('a,'q) bs_pq => bool"

consts
invar :: "('a,'q) bs_pq => bool"
Found termination order: "{}"

Output Query Sledgehammer Symbols
110.62 (3026/6136) (isabelle,isabelle.UTF-8-isabelle)tmr o UG 945/1282MB 1:05 PM

```

```

Isabelle2017 - ex11_tmpl.thy
File Edit Search Markers Folding View Utilities Macros Plugins Help
ex11_tmpl.thy (~/lehre/FDS/SS18/public/exercises/)
101 fun mset :: "('a,'q) bs_pq => 'a multiset"
102   where
103     "mset Empty = {#}"
104     | "mset (Heap _ q) = orig_mset q"
105
106
107 fun invar :: "('a,'q) bs_pq => bool"
108   where
109     "invar Empty <=> True"
110     | "invar (Heap a q) <=> orig_invar q & a ∈ #orig_mset q & a = Min_mset (orig_mset q)"
111
112
113 definition empty :: "('a,'q) bs_pq"
114   where "empty = undefined"
115
116 fun is_empty :: "('a,'q) bs_pq => bool"

consts
invar :: "('a,'q) bs_pq => bool"
Found termination order: "{}"

Output Query Sledgehammer Symbols
110.67 (3031/6136) (isabelle,isabelle.UTF-8-isabelle)tmr o UG 947/1282MB 1:05 PM

```

```

Isabelle2017 - ex11_tmpl.thy (modified)
File Edit Search Markers Folding View Utilities Macros Plugins Help
ex11_tmpl.thy (~/lehre/FDS/SS18/public/exercises/)
101 fun mset :: "('a,'q) bs_pq => 'a multiset"
102   where
103     "mset Empty = {#}"
104     | "mset (Heap _ q) = orig_mset q"
105
106
107 thm orig_mset_get_min .
108
109 fun invar :: "('a,'q) bs_pq => bool"
110   where
111     "invar Empty <=> True"
112     | "invar (Heap a q) <=> orig_invar q & a ∈ #orig_mset q & a = Min_mset (orig_mset q)"
113
114
115 definition empty :: "('a,'q) bs_pq"
116   where "empty = undefined"

[orig_invar ?q; orig_mset ?q ≠ {#}] ==> orig_get_min ?q = Min_mset (orig_mset ?q)

Output Query Sledgehammer Symbols
107.12 (2903/6167) Input/output complete (isabelle,isabelle.UTF-8-isabelle)tmr o UG 947/1282MB 1:06 PM

```

```

Isabelle2017 - ex11_tmpl.thy (modified)
File Edit Search Markers Folding View Utilities Macros Plugins Help
ex11_tmpl.thy (~/lehre/FDS/SS18/public/exercises/)
101 fun mset :: "('a,'q) bs_pq => 'a multiset"
102   where
103     "mset Empty = {#}"
104     | "mset (Heap _ q) = orig_mset q"
105
106
107 thm orig.mset_get_min.
108
109 fun invar :: "('a,'q) bs_pq => bool"
110   where
111     "invar Empty <=> True"
112     | "invar (Heap a q) <=> orig_invar q & a<#orig_mset q & a=Min_mset (orig_mset q)"
113
114
115 definition empty :: "('a,'q) bs_pq"
116   where "empty = undefined"

```

107.12 (2903/6167) (isabelle.isabelle.UTF-8-isabelle)tmr o UG 898/1 282MB 1:07 PM

```

orig_invar ?q; orig_mset ?q ≠ {#} ⇒ orig_get_min ?q = Min_mset (orig_mset ?q)

```

13:07:09

```

Isabelle2017 - ex11_tmpl.thy (modified)
File Edit Search Markers Folding View Utilities Macros Plugins Help
ex11_tmpl.thy (~/lehre/FDS/SS18/public/exercises/)
101 fun mset :: "('a,'q) bs_pq => 'a multiset"
102   where
103     "mset Empty = {#}"
104     | "mset (Heap _ q) = orig_mset q"
105
106
107 thm orig.mset_get_min.
108
109 fun invar :: "('a,'q) bs_pq => bool"
110   where
111     "invar Empty <=> True"
112     | "invar (Heap a q) <=> orig_invar q & a<#orig_mset q & a=Min_mset (orig_mset q)"
113
114
115 definition empty :: "('a,'q) bs_pq"
116   where "empty = undefined"

```

106.1 (2891/6167) (isabelle.isabelle.UTF-8-isabelle)tmr o UG 898/1 282MB 1:07 PM

```

orig_invar ?q; orig_mset ?q ≠ {#} ⇒ orig_get_min ?q = Min_mset (orig_mset ?q)

```

13:07:10

```

Isabelle2017 - ex11_tmpl.thy (modified)
File Edit Search Markers Folding View Utilities Macros Plugins Help
ex11_tmpl.thy (~/lehre/FDS/SS18/public/exercises/)
109 fun invar :: "('a,'q) bs_pq => bool"
110   where
111     "invar Empty <=> True"
112     | "invar (Heap a q) <=> orig_invar q & a<#orig_mset q & a=Min_mset (orig_mset q)"
113
114
115 definition empty :: "('a,'q) bs_pq"
116   where "empty = Empty"
117
118 fun is_empty :: "('a,'q) bs_pq => bool"
119   where
120     "is_empty _ <=> Empt"
121
122 fun insert :: "'a => ('a,'q) bs_pq => ('a,'q) bs_pq"
123   where
124     "insert = undefined"

```

120.23 (3212/6152) (isabelle.isabelle.UTF-8-isabelle)tmr o UG 898/1 282MB 1:07 PM

```

consts
is_empty :: "('a,'q) bs_pq => bool"
Found termination order: "{}"

```

13:07:46

```

Isabelle2017 - ex11_tmpl.thy (modified)
File Edit Search Markers Folding View Utilities Macros Plugins Help
ex11_tmpl.thy (~/lehre/FDS/SS18/public/exercises/)
114
115 definition empty :: "('a,'q) bs_pq"
116   where "empty = Empty"
117
118 fun is_empty :: "('a,'q) bs_pq => bool"
119   where
120     "is_empty bs <=> bs=Empty"
121
122 fun insert :: "'a => ('a,'q) bs_pq => ('a,'q) bs_pq"
123   where
124     "insert x Empty = "
125
126 fun get_min :: "('a,'q) bs_pq => 'a"
127   where
128     "get_min _ = undefined"
129

```

124.23 (3305/6152) Input/output complete (isabelle.isabelle.UTF-8-isabelle)tmr o UG 8/1 257MB 1:08 PM

```

Inner syntax error: unexpected end of input
Failed to parse prop

```

13:08:33

Isabelle2017 - ex11_tmpl.thy (modified)

```
114 definition empty :: "('a,'q) bs_pq"
115   where "empty = Empty"
116
117
118 fun is_empty :: "('a,'q) bs_pq ⇒ bool"
119   where
120     "is_empty bs ←→ bs=Empty"
121
122
123 fun insert :: "'a ⇒ ('a,'q) bs_pq ⇒ ('a,'q) bs_pq"
124   where
125     "insert x Empty = "
126
127 fun get_min :: "('a,'q) bs_pq ⇒ 'a"
128   where
129     "get_min _ = undefined"
```

Inner syntax error: unexpected end of input
Failed to parse prop

120.21 (3210/6152) (isabelle,isabelle.UTF-8-isabelle)tmr o UG 657/1257MB 1:08 PM

Isabelle2017 - ex11_tmpl.thy (modified)

```
114 definition empty :: "('a,'q) bs_pq"
115   where "empty = Empty"
116
117
118 fun is_empty :: "('a,'q) bs_pq ⇒ bool"
119   where
120     "is_empty Empty ←→ True"
121     | "is_empty _ ←→ False"
122
123 fun insert :: "'a ⇒ ('a,'q) bs_pq ⇒ ('a,'q) bs_pq"
124   where
125     "insert x Empty = Heap"
126
127 fun get_min :: "('a,'q) bs_pq ⇒ 'a"
128   where
129     "get_min = undefined"
```

Inner syntax error: unexpected end of input
Failed to parse prop

125.27 (3333/6180) (isabelle,isabelle.UTF-8-isabelle)tmr o UG 657/1257MB 1:09 PM

Isabelle2017 - ex11_tmpl.thy (modified)

```
116   where "empty = Empty"
117
118 fun is_empty :: "('a,'q) bs_pq ⇒ bool"
119   where
120     "is_empty Empty ←→ True"
121     | "is_empty _ ←→ False"
122
123 fun insert :: "'a ⇒ ('a,'q) bs_pq ⇒ ('a,'q) bs_pq"
124   where
125     "insert x Empty = Heap x (orig_insert x orig_empty)"
126     | "insert x (Heap y q) = (())"
127
128 fun get_min :: "('a,'q) bs_pq ⇒ 'a"
129   where
130     "get_min _ = undefined"
```

Inner syntax error: unexpected end of input
Failed to parse prop

126.30 (3393/6240) (isabelle,isabelle.UTF-8-isabelle)tmr o UG 657/1235MB 1:10 PM

Isabelle2017 - ex11_tmpl.thy (modified)

```
126 | "insert x (Heap y q) = Heap (min x y) (orig_insert x q)"
127
128 fun get_min :: "('a,'q) bs_pq ⇒ 'a"
129   where
130     "get_min (Heap x q) = x"
131     | "get_min Empty = undefined"
132
133 fun del_min :: "('a,'q) bs_pq ⇒ ('a,'q) bs_pq"
134   where
135     "del_min Empty = undefined"
136     | "del_min (Heap x q) = (())"
137
138
139 Lemmas [simp] = orig.is_empty orig.mset_get_min orig.mset_del_min
140               orig.mset_insert orig.mset_empty
141               orig.invar empty orig.invar insert orig.invar del_min
```

Malformed command syntax

136.29 (3655/6337) (isabelle,isabelle.UTF-8-isabelle)tmr o UG 75/1196MB 1:14 PM

```

127 fun get_min :: "('a, 'q) bs_pq => 'a"
128   where
129     "get_min (Heap x q) = x"
130   | "get_min Empty = undefined"
131
132
133 fun del_min :: "('a, 'q) bs_pq => ('a, 'q) bs_pq"
134   where
135     "del_min Empty = undefined"
136   | "del_min (Heap x q) = (let q = orig_del_min q in undefined)"
137
138
139 lemmas [simp] = orig.is_empty orig.mset_get_min orig.mset_del_min
140               orig.mset_insert orig.mset_empty
141               orig.invar_empty orig.invar_insert orig.invar_del_min
142

```

consts
del_min :: "('a, 'q) bs_pq => ('a, 'q) bs_pq"
Found termination order: "{}"

```

127 fun get_min :: "('a, 'q) bs_pq => 'a"
128   where
129     "get_min (Heap x q) = x"
130   | "get_min Empty = undefined"
131
132
133 fun del_min :: "('a, 'q) bs_pq => ('a, 'q) bs_pq"
134   where
135     "del_min Empty = undefined"
136   | "del_min (Heap x q) = (let q = orig_del_min q in Heap (orig_get_min q) q)"
137
138
139 lemmas [simp] = orig.is_empty orig.mset_get_min orig.mset_del_min
140               orig.mset_insert orig.mset_empty
141               orig.invar_empty orig.invar_insert orig.invar_del_min
142

```

consts
del_min :: "('a, 'q) bs_pq => ('a, 'q) bs_pq"
Found termination order: "{}"

```

130 "get_min (Heap x q) = x"
131 | "get_min Empty = undefined"
132
133 fun del_min :: "('a, 'q) bs_pq => ('a, 'q) bs_pq"
134   where
135     "del_min Empty = undefined"
136   | "del_min (Heap x q) = (let
137     q = orig_del_min q
138   in
139     if orig_is_empty q then Empty
140     else Heap (orig_get_min q) q)"
141
142
143 lemmas [simp] = orig.is_empty orig.mset_get_min orig.mset_del_min
144               orig.mset_insert orig.mset_empty
145               orig.invar_empty orig.invar_insert orig.invar_del_min
146

```

consts
del_min :: "('a, 'q) bs_pq => ('a, 'q) bs_pq"
Found termination order: "{}"

```

136 | "del_min (Heap x q) = (let
137   q = orig_del_min q
138 in
139   if orig_is_empty q then Empty
140   else Heap (orig_get_min q) q)"
141
142
143 lemmas [simp] = orig.is_empty orig.mset_get_min orig.mset_del_min
144               orig.mset_insert orig.mset_empty
145               orig.invar_empty orig.invar_insert orig.invar_del_min
146
147 text <Show that your new implementation satisfies the priority queue interface!>
148 sublocale Priority_Queue empty is_empty insert get_min del_min invar mset
149   apply unfold_locales
150   proof goal_cases
151     case 1

```

theorem
• orig_invar ?q => orig_is_empty ?q = (orig_mset ?q = {#})
• [orig_invar ?q; orig_mset ?q ≠ {#}] => orig_get_min ?q = Min mset (orig_mset ?q)

```

Isabelle2017 - Priority_Queue.thy
File Edit Search Markers Folding View Utilities Macros Plugins Help
Priority_Queue.thy (ISABELLE_HOME/src/HOL/Data_Structures)
18 and mset :: "'a multiset"
19 assumes mset_empty: "mset empty = {#}"
20 and is_empty: "invar q  $\implies$  is_empty q = (mset q = {#})"
21 and mset_insert: "invar q  $\implies$  mset (insert x q) = mset q + {#x#}"
22 and mset_del_min: "invar q  $\implies$  mset q  $\neq$  {#}  $\implies$ 
23   mset (del_min q) = mset q - {# get_min q #}"
24 and mset_get_min: "invar q  $\implies$  mset q  $\neq$  {#}  $\implies$  get_min q = Min_mset (mset q)"
25 and invar_empty: "invar empty"
26 and invar_insert: "invar q  $\implies$  invar (insert x q)"
27 and invar_del_min: "invar q  $\implies$  mset q  $\neq$  {#}  $\implies$  invar (del_min q)"
28
29 text <Extend locale with <merge>. Need to enforce that '<q>' is the same in both locales.>
30
31 locale Priority_Queue_Merge = Priority_Queue where empty = empty for empty :: 'a +
32 fixes merge :: "'a + 'a + 'a"
33 assumes mset_merge: "[| invar q1; invar q2 |]  $\implies$  mset (merge q1 q2) = mset q1 + mset q2"

```

Proof state Auto update Update Search: 100%

locale

Priority Queue

Output Query Sledgehammer Symbols

21.5 (492/1225) (isabelle.isabelle.UTF-8-isabelle)tmr o UG 13:41:155MB 1:17 PM

```

Isabelle2017 - ex11_tmpl.thy
File Edit Search Markers Folding View Utilities Macros Plugins Help
ex11_tmpl.thy (~/lehre/FDS/SS18/public/exercises/)
145 orig.invar_empty orig.invar_insert orig.invar_del_min
146
147 text <Show that your new implementation satisfies the priority queue interface!>
148 sublocale Priority_Queue empty is_empty insert get_min del_min invar mset
149 apply unfold_locales
150 proof goal_cases
151 case 1
152 then show ?case sorry
153 next
154 case (2 q) -- <and so on>
155 oops

```

Proof state Auto update Update Search: 100%

```

proof (state)
goal (8 subgoals):
1. local.mset local.empty = {#}
2.  $\wedge q$ . invar q  $\implies$  is_empty q = (local.mset q = {#})
3.  $\wedge q x$ . invar q  $\implies$  local.mset (local.insert x q) = local.mset q + {#x#}
4.  $\wedge q$ . [invar q; local.mset q  $\neq$  {#}]  $\implies$  local.mset (local.del_min q) = local.mset q - {#local.get_min q#}
5.  $\wedge q$ . [invar q; local.mset q  $\neq$  {#}]  $\implies$  local.get_min q = Min_mset (local.mset q)
6. invar local.empty

```

Output Query Sledgehammer Symbols

150.19 (4139/6451) (isabelle.isabelle.UTF-8-isabelle)tmr o UG 13:41:140MB 1:19 PM

```

Isabelle2017 - ex11_tmpl.thy (modified)
File Edit Search Markers Folding View Utilities Macros Plugins Help
ex11_tmpl.thy (~/lehre/FDS/SS18/public/exercises/)
147 text <Show that your new implementation satisfies the priority queue interface!>
148 sublocale Priority_Queue empty is_empty insert get_min del_min invar mset
149 apply unfold_locales
150 proof goal_cases
151 case 1
152 then show ?case.
153
154 next
155 case (2 q) -- <and so on>
156 oops
157

```

Proof state Auto update Update Search: 100%

```

proof (prove)
goal (1 subgoal):
1. local.mset local.empty = {#}

```

Output Query Sledgehammer Symbols

153.7 (4179/6453) (isabelle.isabelle.UTF-8-isabelle)tmr o UG 13:21:140MB 1:20 PM

```

Isabelle2017 - ex11_tmpl.thy
File Edit Search Markers Folding View Utilities Macros Plugins Help
ex11_tmpl.thy (~/lehre/FDS/SS18/public/exercises/)
149 apply unfold_locales
150 proof goal_cases
151 case 1
152 then show ?case
153   unfolding empty_def by simp
154 next
155 case (2 q) then show ?case
156   apply auto
157
158
159 end

```

Proof state Auto update Update Search: 100%

```

proof (prove)
goal (2 subgoals):
1. [invar q; is_empty q]  $\implies$  local.mset q = {#}
2. [invar q; local.mset q = {#}]  $\implies$  is_empty q

```

Output Query Sledgehammer Symbols

155.18 (4229/6488) (isabelle.isabelle.UTF-8-isabelle)tmr o UG 13:21:123MB 1:21 PM

```

Isabelle2017 - ex11_tmpl.thy
File Edit Search Markers Folding View Utilities Macros Plugins Help
ex11_tmpl.thy (~/lehre/FDS/SS18/public/exercises/)
149 apply unfold_locales
150 proof goal_cases
151 case 1
152 then show ?case
153 unfolding empty_def by simp
154 next
155 case (2 q) then show ?case
156 apply auto
157
158 end

proof (prove)
goal (2 subgoals):
1. [invar q; is_empty q] ==> local.mset q = {#}
2. [invar q; local.mset q = {#}] ==> is_empty q

```

```

Isabelle2017 - ex11_tmpl.thy
File Edit Search Markers Folding View Utilities Macros Plugins Help
ex11_tmpl.thy (~/lehre/FDS/SS18/public/exercises/)
103 "mset Empty = {#}"
104 | "mset (Heap _ q) = orig_mset q"
105
106
107 thm orig_mset_get_min
108
109 fun invar :: "('a, 'q) bs_pq ==> bool"
110 where
111 "invar Empty ==> True"
112 | "invar (Heap a q) ==> orig_invar q ^ a <# orig_mset q ^ a = Min_mset (orig_mset q)"
113
const
invar :: "('a, 'q) bs_pq ==> bool"
Found termination order: "{}"

```

```

Isabelle2017 - ex11_tmpl.thy
File Edit Search Markers Folding View Utilities Macros Plugins Help
ex11_tmpl.thy (~/lehre/FDS/SS18/public/exercises/)
150 proof goal_cases
151 case 1
152 then show ?case
153 unfolding empty_def by simp
154 next
155 case (2 q) then show ?case
156 apply auto
157
158 end
159
160

proof (prove)
goal (2 subgoals):
1. [invar q; is_empty q] ==> local.mset q = {#}
2. [invar q; local.mset q = {#}] ==> is_empty q

```

```

Isabelle2017 - ex11_tmpl.thy
File Edit Search Markers Folding View Utilities Macros Plugins Help
ex11_tmpl.thy (~/lehre/FDS/SS18/public/exercises/)
166 next
167 case 6
168 then show ?case unfolding empty_def by auto
169 next
170 case (7 q x)
171 then show ?case
172 apply (cases q)
173 apply (auto)
174 next
175 case (8 q)
176 then show ?case by (cases q) auto

proof (prove)
using this:
invar q

goal (1 subgoal):
1. invar (local.insert x q)

```

Isabelle2017 - ex11_tmpl.thy (modified)

```

167 case 6
168 then show ?case unfolding empty_def by auto
169 next
170 case (7 q x)
171 then show ?case
172 apply (cases q)
173 apply (auto simp: min_def)
174 .....
175 next
176 case (8 q)
177 then show ?case by (cases q) auto

```

proof (prove)
goal (2 subgoals):

- $\bigwedge x22. [q = \text{Heap} (\text{Min_mset} (\text{orig_mset } x22)) x22; \text{orig_invar } x22; \text{Min_mset} (\text{orig_mset } x22) \in\# \text{orig_mset } x22; x \leq \text{Min_mset} (\text{orig_mset } x22)] \Rightarrow x = \text{Min} (\text{Set.insert } x (\text{set_mset } (\text{orig_mset } x22)))$
- $\bigwedge x22. [q = \text{Heap} (\text{Min_mset} (\text{orig_mset } x22)) x22; \text{orig_invar } x22; \text{Min_mset} (\text{orig_mset } x22) \in\# \text{orig_mset } x22; \neg x \leq \text{Min_mset} (\text{orig_mset } x22)] \Rightarrow \text{Min_mset} (\text{orig_mset } x22) = \text{Min} (\text{Set.insert } x (\text{set_mset } (\text{orig_mset } x22)))$

173.32 (4612/6915) Input/output complete (Isabelle, Isabelle, UTF-8-Isabelle) tmr o UG 349/1027MB 1:26 PM

Isabelle2017 - ex11_tmpl.thy (modified)

```

167 case 6
168 then show ?case unfolding empty_def by auto
169 next
170 case (7 q x)
171 then show ?case
172 apply (cases q)
173 apply (auto simp: min_def)
174 .....
175 next
176 case (8 q)
177 then show ?case by (cases q) auto

```

proof (prove)
goal (2 subgoals):

- $\bigwedge x22. [q = \text{Heap} (\text{Min_mset} (\text{orig_mset } x22)) x22; \text{orig_invar } x22; \text{Min_mset} (\text{orig_mset } x22) \in\# \text{orig_mset } x22; x \leq \text{Min_mset} (\text{orig_mset } x22)] \Rightarrow x = \text{Min} (\text{Set.insert } x (\text{set_mset } (\text{orig_mset } x22)))$
- $\bigwedge x22. [q = \text{Heap} (\text{Min_mset} (\text{orig_mset } x22)) x22; \text{orig_invar } x22; \text{Min_mset} (\text{orig_mset } x22) \in\# \text{orig_mset } x22; \neg x \leq \text{Min_mset} (\text{orig_mset } x22)] \Rightarrow \text{Min_mset} (\text{orig_mset } x22) = \text{Min} (\text{Set.insert } x (\text{set_mset } (\text{orig_mset } x22)))$

173.32 (4612/6915) (Isabelle, Isabelle, UTF-8-Isabelle) tmr o UG 366/1027MB 1:27 PM

Isabelle2017 - ex11_tmpl.thy (modified)

```

167 case 6
168 then show ?case unfolding empty_def by auto
169 next
170 case (7 q x)
171 then show ?case
172 apply (cases q)
173 apply (auto simp: min_def)
174 .....
175 next
176 case (8 q)
177 then show ?case by (cases q) auto

```

proof (prove)
goal (2 subgoals):

- $\bigwedge x22. [q = \text{Heap} (\text{Min_mset} (\text{orig_mset } x22)) x22; \text{orig_invar } x22; \text{Min_mset} (\text{orig_mset } x22) \in\# \text{orig_mset } x22; x \leq \text{Min_mset} (\text{orig_mset } x22)] \Rightarrow x = \text{Min} (\text{Set.insert } x (\text{set_mset } (\text{orig_mset } x22)))$
- $\bigwedge x22. [q = \text{Heap} (\text{Min_mset} (\text{orig_mset } x22)) x22; \text{orig_invar } x22; \text{Min_mset} (\text{orig_mset } x22) \in\# \text{orig_mset } x22; \neg x \leq \text{Min_mset} (\text{orig_mset } x22)] \Rightarrow \text{Min_mset} (\text{orig_mset } x22) = \text{Min} (\text{Set.insert } x (\text{set_mset } (\text{orig_mset } x22)))$

173.33 (4613/6915) (Isabelle, Isabelle, UTF-8-Isabelle) tmr o UG 383/1027MB 1:27 PM

Isabelle2017 - ex11_tmpl.thy

```

167 case 6
168 then show ?case unfolding empty_def by auto
169 next
170 case (7 q x)
171 then show ?case
172 apply (cases q)
173 apply (auto simp: min_def)
174 sledgehammer
175 .....
176 next
177 case (8 q)

```

Sledgehammering...

174.19 (4632/6928) (Isabelle, Isabelle, UTF-8-Isabelle) tmr o UG 383/1027MB 1:27 PM

Isabelle2017 - ex11_tmpl.thy

```

167 case 6
168 then show ?case unfolding empty_def by auto
169 next
170 case (7 q x)
171 then show ?case
172 apply (cases q)
173 apply (auto simp: min_def)

```

find_theorems

"Min"
"Set.insert"

found 8 theorem(s):

- Lattices_Big.linorder_class.Min_singleton: $\text{Min } \{?x\} = ?x$
- Lattices_Big.linorder_class.Min_eq_fold: $\text{finite } ?A \implies \text{Min } (\text{Set.insert } ?x ?A) = \text{Finite_Set.fold min } ?x ?A$
- Lattices_Big.linorder_class.Min_insert: $[\text{finite } ?A; ?A \neq \{\}] \implies \text{Min } (\text{Set.insert } ?x ?A) = \text{min } ?x (\text{Min } ?A)$
- Lattices_Big.linorder_class.Min_insert2: $[\text{finite } ?A; \bigwedge b. b \in ?A \implies ?a \leq b] \implies \text{Min } (\text{Set.insert } ?a ?A) = ?a$
- Lattices_Big.linorder_class.Min_insert_not_elem: $[\text{finite } ?A; \bigwedge b \in ?A. b \neq ?a] \implies \text{Min } (\text{Set.insert } ?a ?A) = \text{Min } ?A$

174.29 (4642/6944) (isabelle,isabelle,UTF-8-isabelle)tmr o UG 113/1005MB 1:28 PM

Isabelle2017 - ex11_tmpl.thy

```

167 case 6
168 then show ?case unfolding empty_def by auto
169 next
170 case (7 q x)
171 then show ?case
172 apply (cases q)
173 apply (auto simp: min_def)

```

find_theorems

"Min"
"Set.insert"

found 8 theorem(s):

- Lattices_Big.linorder_class.Min_singleton: $\text{Min } \{?x\} = ?x$
- Lattices_Big.linorder_class.Min_eq_fold: $\text{finite } ?A \implies \text{Min } (\text{Set.insert } ?x ?A) = \text{Finite_Set.fold min } ?x ?A$
- Lattices_Big.linorder_class.Min_insert: $[\text{finite } ?A; ?A \neq \{\}] \implies \text{Min } (\text{Set.insert } ?x ?A) = \text{min } ?x (\text{Min } ?A)$
- Lattices_Big.linorder_class.Min_insert2: $[\text{finite } ?A; \bigwedge b. b \in ?A \implies ?a \leq b] \implies \text{Min } (\text{Set.insert } ?a ?A) = ?a$
- Lattices_Big.linorder_class.Min_insert_not_elem: $[\text{finite } ?A; \bigwedge b \in ?A. b \neq ?a] \implies \text{Min } (\text{Set.insert } ?a ?A) = \text{Min } ?A$

174.29 (4642/6944) (isabelle,isabelle,UTF-8-isabelle)tmr o UG 126/1005MB 1:28 PM

Isabelle2017 - ex11_tmpl.thy (modified)

```

169 next
170 case (7 q x)
171 then show ?case
172 apply (cases q)
173 apply (auto simp: min_def)
174 apply (rule Min_insert2[symmetric])
175 apply simp

```

find_theorems Min Set.insert

proof (prove)

goal (2 subgoals):

- $\bigwedge x22 b. [q = \text{Heap } (\text{Min_mset } (\text{orig_mset } x22)) x22; \text{orig_invar } x22; \text{Min_mset } (\text{orig_mset } x22) \in \# \text{orig_mset } x22; x \leq \text{Min_mset } (\text{orig_mset } x22); b \in \# \text{orig_mset } x22] \implies x \leq b$
- $\bigwedge x22. [q = \text{Heap } (\text{Min_mset } (\text{orig_mset } x22)) x22; \text{orig_invar } x22;$

176.7 (4679/7016) (isabelle,isabelle,UTF-8-isabelle)tmr o UG 113/9983MB 1:29 PM

Isabelle2017 - ex11_tmpl.thy

```

169 next
170 case (7 q x)
171 then show ?case
172 apply (cases q)
173 apply (auto simp: min_def)
174 apply (rule Min_insert2[symmetric])
175 apply simp

```

sledgehammer

find_theorems Min Set.insert

proof (prove)

goal (2 subgoals):

- $\bigwedge x22 b. [q = \text{Heap } (\text{Min_mset } (\text{orig_mset } x22)) x22; \text{orig_invar } x22; \text{Min_mset } (\text{orig_mset } x22) \in \# \text{orig_mset } x22; x \leq \text{Min_mset } (\text{orig_mset } x22); b \in \# \text{orig_mset } x22] \implies x \leq b$
- $\bigwedge x22. [q = \text{Heap } (\text{Min_mset } (\text{orig_mset } x22)) x22; \text{orig_invar } x22;$

175.17 (4672/7027) (isabelle,isabelle,UTF-8-isabelle)tmr o UG 20/971MB 1:30 PM

```

Isabelle2017 - ex11_tmpl.thy (modified)
File Edit Search Markers Folding View Utilities Magros Plugins Help
ex11_tmpl.thy (~/lehre/FDS/SS18/public/exercises/)
170 case (7 q x)
171 then show ?case
172   apply (cases q)
173   apply (auto simp: min_def)
174   apply (rule Min_insert2[symmetric])
175   apply simp
176   using Min.boundedE apply auto[1]
177   sledgehammer
178
179 next
180 case (8 q)
181 then show ?case by (cases q) auto
182 qed

proof (prove)
goal (1 subgoal):
1.  $\wedge x22. [q = \text{Heap} (\text{Min\_mset} (\text{orig\_mset } x22)) x22; \text{orig\_invar } x22;$ 
    $\text{Min\_mset} (\text{orig\_mset } x22) \in\# \text{orig\_mset } x22; \neg x \leq \text{Min\_mset} (\text{orig\_mset } x22)]$ 
 $\implies \text{Min\_mset} (\text{orig\_mset } x22) = \text{Min} (\text{Set.insert } x (\text{set\_mset } (\text{orig\_mset } x22)))$ 

```

```

Isabelle2017 - ex11_tmpl.thy
File Edit Search Markers Folding View Utilities Magros Plugins Help
ex11_tmpl.thy (~/lehre/FDS/SS18/public/exercises/)
171 then show ?case
172   apply (cases q)
173   apply (auto simp: min_def)
174   apply (rule Min_insert2[symmetric])
175   apply simp
176   using Min.boundedE apply auto[1]
177   sledgehammer
178
179 next
180 case (8 q)
181 then show ?case by (cases q) auto
182 qed

Sledgehammering...
Proof found...

```

```

Isabelle2017 - ex11_tmpl.thy
File Edit Search Markers Folding View Utilities Magros Plugins Help
ex11_tmpl.thy (~/lehre/FDS/SS18/public/exercises/)
171 then show ?case
172   apply (cases q)
173   apply (auto simp: min_def)
174   apply (rule Min_insert2[symmetric])
175   apply simp
176   using Min.boundedE apply auto[1]
177   sledgehammer
178
179 next
180 case (8 q)
181 then show ?case by (cases q) auto
182 qed

Sledgehammering...
Proof found...
"z3": Try this: by (metis Min_insert all_not_in_conv finite_set_mset min_def) (100 ms)
"cvc4": Try this: by (metis Min_insert ball_empty finite_set_mset min_def) (48 ms)
"spass": Try this: by (metis Min_insert all_not_in_conv finite_set_mset min_def) (64 ms)

```

```

Isabelle2017 - ex11_tmpl.thy
File Edit Search Markers Folding View Utilities Magros Plugins Help
ex11_tmpl.thy (~/lehre/FDS/SS18/public/exercises/)
170 case (7 q x)
171 then show ?case
172   apply (cases q)
173   apply (auto simp: min_def)
174   apply (rule Min_insert2[symmetric])
175   apply simp
176   using Min.boundedE apply auto[1]
177   by (metis Min_insert ball_empty finite_set_mset min_def)
178
179 next
180 case (8 q)
181 then show ?case by (cases q) auto
182 qed

proof (prove)
goal (2 subgoals):
1.  $\wedge x22. [q = \text{Heap} (\text{Min\_mset} (\text{orig\_mset } x22)) x22; \text{orig\_invar } x22;$ 
    $\text{Min\_mset} (\text{orig\_mset } x22) \in\# \text{orig\_mset } x22; x \leq \text{Min\_mset} (\text{orig\_mset } x22)]$ 
 $\implies x = \text{Min} (\text{Set.insert } x (\text{set\_mset} (\text{orig\_mset } x22)))$ 
2.  $\wedge x22. [q = \text{Heap} (\text{Min\_mset} (\text{orig\_mset } x22)) x22; \text{orig\_invar } x22;$ 
    $\text{Min\_mset} (\text{orig\_mset } x22) \in\# \text{orig\_mset } x22; \neg x < \text{Min\_mset} (\text{orig\_mset } x22)]$ 

```

```

Isabelle2017 - ex11_tmpl.thy
File Edit Search Markers Folding View Utilities Macros Plugins Help
ex11_tmpl.thy (~/lehre/FDS/SS18/public/exercises/)
142 lemmas [simp] = orig.is_empty orig.mset_get_min orig.mset_del_min
143 orig.mset_insert orig.mset_empty
144 orig.invar_empty orig.invar_insert orig.invar_del_min
145
146 lemma "[m ≠ {#}; x ≤ Min_mset m] ⇒ x = Min (Set.insert x (set_mset m))"
147
148
149
150 text <Show that your new implementation satisfies the priority queue interface!>
151 sublocale Priority_Queue empty is_empty insert get_min del_min invar mset
152 apply unfold_locales
153 proof goal_cases
154
proof (prove)
goal (1 subgoal):
1. [m ≠ {#}; x ≤ Min_mset m] ⇒ x = Min (Set.insert x (set_mset m))

```

```

Isabelle2017 - ex11_tmpl.thy
File Edit Search Markers Folding View Utilities Macros Plugins Help
ex11_tmpl.thy (~/lehre/FDS/SS18/public/exercises/)
145 orig.invar_empty orig.invar_insert orig.invar_del_min
146
147 lemma "[m ≠ {#}; x ≤ Min_mset m] ⇒ x = Min (Set.insert x (set_mset m))"
148
149
150
151 text <Show that your new implementation satisfies the priority queue interface!>
152 sublocale Priority_Queue empty is_empty insert get_min del_min invar mset
153 apply unfold_locales
154 proof goal_cases
155 case 1
156 then show ?case
157
Sledgehammering...

```

```

Isabelle2017 - ex11_tmpl.thy
File Edit Search Markers Folding View Utilities Macros Plugins Help
ex11_tmpl.thy (~/lehre/FDS/SS18/public/exercises/)
172 then show ?case unfolding empty_def by auto
173 next
174 case (7 q x)
175 then show ?case
176 apply (cases q)
177 apply (auto simp: min_def aux1)
178 apply (rule aux1)
179 apply auto []
180 apply auto []
181
182
183
184
185
186
187
188
189 apply (rule Min_insert2[symmetric])
190
proof (prove)
goal (3 subgoals):
1.  $\wedge x22. [q = \text{Heap} (\text{Min\_mset} (\text{orig\_mset } x22)) \ x22; \text{orig\_invar } x22; \text{Min\_mset} (\text{orig\_mset } x22) \in \# \text{orig\_mset } x22; x \leq \text{Min\_mset} (\text{orig\_mset } x22)] \Rightarrow \text{orig\_mset } x22 \neq \{ \# \}$ 
2.  $\wedge x22. [q = \text{Heap} (\text{Min\_mset} (\text{orig\_mset } x22)) \ x22; \text{orig\_invar } x22; \text{Min\_mset} (\text{orig\_mset } x22) \in \# \text{orig\_mset } x22; x < \text{Min\_mset} (\text{orig\_mset } x22)]$ 

```

```

Isabelle2017 - ex11_tmpl.thy
File Edit Search Markers Folding View Utilities Macros Plugins Help
ex11_tmpl.thy (~/lehre/FDS/SS18/public/exercises/)
167 next
168 case (4 q)
169 then show ?case by (cases q) auto
170 next
171 case (5 q)
172 then show ?case by (cases q) auto
173 next
174 case 6
175 then show ?case unfolding empty_def by auto
176 next
177 case (7 q x)
178 then show ?case
179
proof (prove)
goal (2 subgoals):
1. [invar q; q = Empty] ⇒ invar (local.insert x q)
2.  $\wedge x21 \ x22. [\text{invar } q; q = \text{Heap } x21 \ x22] \Rightarrow \text{invar} (\text{local.insert } x \ q)$ 

```



```

Isabelle2017 - ex11_tmpl.thy
File Edit Search Markers Folding View Utilities Macros Plugins Help
ex11_tmpl.thy (~/lehre/FDS/SS18/public/exercises/)
147 lemma aux1: "m ≠ {#}; x ≤ Min_mset m ⇒ x = Min (Set.insert x (set_mset m))"
148 by (metis Min_insert2 Min_le finite_set_mset order_trans)
149
150
151 lemma [simp]: "Min_mset m ∈ # m ↔ m ≠ {#}" by auto
152
153
154 text <Show that your new implementation satisfies the priority queue interface!>
155 sublocale Priority_Queue empty is_empty insert get_min del_min invar mset
156 apply unfold_locales
157 proof goal_cases
158 case 1

```

theorem (Min_mset ?m ∈ # ?m) = (?m ≠ {#})

153.1 (4131/7331) (Isabelle, Isabelle, UTF-8-Isabelle) tmr o UG 07/819MB 1:40 PM

```

Isabelle2017 - ex11_tmpl.thy
File Edit Search Markers Folding View Utilities Macros Plugins Help
ex11_tmpl.thy (~/lehre/FDS/SS18/public/exercises/)
170 next
171 case (5 q)
172 then show ?case by (cases q) auto
173 next
174 case 6
175 then show ?case unfolding empty_def by auto
176 next
177 case (7 q x)
178 then show ?case
179 by (cases q) (auto simp: min_def)
180
181 next
182 case (8 q)
183 then show ?case by (cases q) auto
184 qed
185
186

```

proof (state)

177.1 (4717/7089) (Isabelle, Isabelle, UTF-8-Isabelle) tmr o UG 09/803MB 1:41 PM

```

Isabelle2017 - ex11_tmpl.thy
File Edit Search Markers Folding View Utilities Macros Plugins Help
ex11_tmpl.thy (~/lehre/FDS/SS18/public/exercises/)
182 case (8 q)
183 then show ?case by (cases q) auto
184 qed
185
186
187 end
188
189 text <
190 \Homework{Heap}{29.-6.-2018}
191
192 A binary tree can be encoded as an array <a1, ..., an>, such that
193 the parent of node <ai> is node <a(i div 2)>.
194
195 Thus, for a heap, each node is greater than or equal to its parent:
196 >
197
198 definition parent :: "nat ⇒ nat" where "parent i ≡ (i+1) div 2 - 1"

```

192.13 (4916/7089) (Isabelle, Isabelle, UTF-8-Isabelle) tmr o UG 03/797MB 1:42 PM

```

Isabelle2017 - ex11_tmpl.thy
File Edit Search Markers Folding View Utilities Macros Plugins Help
ex11_tmpl.thy (~/lehre/FDS/SS18/public/exercises/)
182 case (8 q)
183 then show ?case by (cases q) auto
184 qed
185
186
187 end
188
189 text <
190 \Homework{Heap}{29.-6.-2018}
191
192 A binary tree can be encoded as an array <a1, ..., an>, such that
193 the parent of node <ai> is node <a(i div 2)>.
194
195 Thus, for a heap, each node is greater than or equal to its parent:
196 >
197
198 definition parent :: "nat ⇒ nat" where "parent i ≡ (i+1) div 2 - 1"

```

193.1 (4972/7089) (Isabelle, Isabelle, UTF-8-Isabelle) tmr o UG 30/793MB 1:42 PM

```

188 text <
189 \Homework{Heap}{29.-6.-2018}
190
191
192 A binary tree can be encoded as an array <[a1, ..., an]>, such that
193 the parent of node <ai> is node <a(i div 2)>.
194
195 Thus, for a heap, each node is greater than or equal to its parent:
196
197
198 definition parent :: "nat ⇒ nat" where "parent i ≡ (i+1) div 2 - 1"
199 definition is_heap :: "'a::linorder list ⇒ bool"
200 where "is_heap h ≡ ∀i<length h. h!i ≥ h!parent i"
201
202
203 text <A heap with a single defect at index <j> is characterized as follows:
204 The heap property holds for all elements except <j>,
  
```

```

191 A binary tree can be encoded as an array <[a1, ..., an]>, such that
192 the parent of node <ai> is node <a(i div 2)>.
193
194 Thus, for a heap, each node is greater than or equal to its parent:
195
196
197
198 definition parent :: "nat ⇒ nat" where "parent i ≡ (i+1) div 2 - 1"
199 definition is_heap :: "'a::linorder list ⇒ bool"
200 where "is_heap h ≡ ∀i<length h. h!i ≥ h!parent i"
201
202
203 text <A heap with a single defect at index <j> is characterized as follows:
204 The heap property holds for all elements except <j>,
205 and the children of <j> must also be greater than their grand-parent.
206
207 definition is_heap_except :: "nat ⇒ 'a::linorder list ⇒ bool" where
  
```

```

198 definition parent :: "nat ⇒ nat" where "parent i ≡ (i+1) div 2 - 1"
199 definition is_heap :: "'a::linorder list ⇒ bool"
200 where "is_heap h ≡ ∀i<length h. h!i ≥ h!parent i"
201
202
203 text <A heap with a single defect at index <j> is characterized as follows:
204 The heap property holds for all elements except <j>,
205 and the children of <j> must also be greater than their grand-parent.
206
207 definition is_heap_except :: "nat ⇒ 'a::linorder list ⇒ bool" where
208 "is_heap_except j h ≡
209 (∀i<length h. i≠j → h!i ≥ h!(parent i))
210 ∧ (∀i<length h. parent i = j → h!i ≥ h!(parent j))"
211
212
213 text <
214
  
```

```

230 else sift_up (swap i (parent i) h) (parent i))"
231
232
233 text <Show that @{const sift_up} restores the heap
234 and preserves the multiset of elements in the heap>
235
236 lemma sift_up_restore_heap:
237 "is_heap_except j h ⇒ j<length h ⇒ is_heap (sift_up h j)"
238 sorry
239
240 lemma sift_up_mset: "j<length h ⇒ mset (sift_up h j) = mset h"
241 sorry
242
243
244 text <For \textbf{3 bonus points}, add an empty, insert, and get-min
245 function to the heap implementation, and prove their essential properties.
246
  
```

```

Isabelle2017 - ex11_tmpl.thy
File Edit Search Markers Folding View Utilities Macros Plugins Help
ex11_tmpl.thy (~/lehre/FDS/SS18/public/exercises/)
240
241 Lemma sift_up_mset: "j < length h  $\implies$  mset (sift_up h j) = mset h"
242 sorry
243
244 text <For \textbf{3} bonus points>, add an empty, insert, and get-min
245 function to the heap implementation, and prove their essential properties.
246
247
248 definition emp :: "'a::linorder list" where "emp  $\equiv$  undefined"
249 definition get_min :: "'a::linorder list  $\Rightarrow$  'a" where "get_min h  $\equiv$  undefined"
250 definition ins :: "'a::linorder  $\Rightarrow$  'a list  $\Rightarrow$  'a list" where "ins x h = undefined"
251
252 Lemma invar_empty: "is_heap emp" sorry
253
254 Lemma mset_empty: "mset emp = {#}" sorry
255
256 Lemma mset_get_min: "is_heap h  $\implies$  mset h  $\neq$  {#}  $\implies$  get_min h = Min_mset (mset h)" sorry

```

251.1 (6790/7089) (isabelle.isabelle,UTF-8-isabelle)tmr c UG 275/775MB 1:50 PM
 13:50:47

```

Isabelle2017 - ex11_tmpl.thy
File Edit Search Markers Folding View Utilities Macros Plugins Help
ex11_tmpl.thy (~/lehre/FDS/SS18/public/exercises/)
246
247
248 definition emp :: "'a::linorder list" where "emp  $\equiv$  undefined"
249 definition get_min :: "'a::linorder list  $\Rightarrow$  'a" where "get_min h  $\equiv$  undefined"
250 definition ins :: "'a::linorder  $\Rightarrow$  'a list  $\Rightarrow$  'a list" where "ins x h = undefined"
251
252 Lemma invar_empty: "is_heap emp" sorry
253
254 Lemma mset_empty: "mset emp = {#}" sorry
255
256 Lemma mset_get_min: "is_heap h  $\implies$  mset h  $\neq$  {#}  $\implies$  get_min h = Min_mset (mset h)" sorry
257
258 Lemma invar_ins: "is_heap h  $\implies$  is_heap (ins x h)" sorry
259
260 Lemma mset_ins: "mset (ins x h) = mset h + {#x#}" sorry

```

251.1 (6830/7089) (isabelle.isabelle,UTF-8-isabelle)tmr c UG 245/772MB 1:51 PM
 13:51:22

```

Isabelle2017 - ex11_tmpl.thy
File Edit Search Markers Folding View Utilities Macros Plugins Help
ex11_tmpl.thy (~/lehre/FDS/SS18/public/exercises/)
233
234 text <Show that @const sift_up> restores the heap
235 and preserves the multiset of elements in the heap>
236
237 Lemma sift_up_restore_heap:
238 "is_heap_except j h  $\implies$  j < length h  $\implies$  is_heap (sift_up h j)"
239 sorry
240
241 Lemma sift_up_mset: "j < length h  $\implies$  mset (sift_up h j) = mset h"
242 sorry
243
244 text <For \textbf{3} bonus points>, add an empty, insert, and get-min
245 function to the heap implementation, and prove their essential properties.
246
247
248 definition emp :: "'a::linorder list" where "emp  $\equiv$  undefined"
249 definition get_min :: "'a::linorder list  $\Rightarrow$  'a" where "get_min h  $\equiv$  undefined"

```

239.1 (6347/7089) (isabelle.isabelle,UTF-8-isabelle)tmr c UG 6/764MB 1:51 PM
 13:51:49