Script generated by TTT

Title: Mayr: 2011 ds (27.01.2012)

Date: Fri Jan 27 16:02:42 CET 2012

Duration: 86:15 min

Pages: 36

WS 2011

Diskrete Strukturen

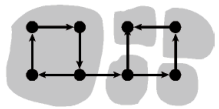
Ernst W. Mayr

Fakultät für Informatik
TU München

http://www14.in.tum.de/lehre/2011WS/ds/

Wintersemester 2011

Beispiel 298

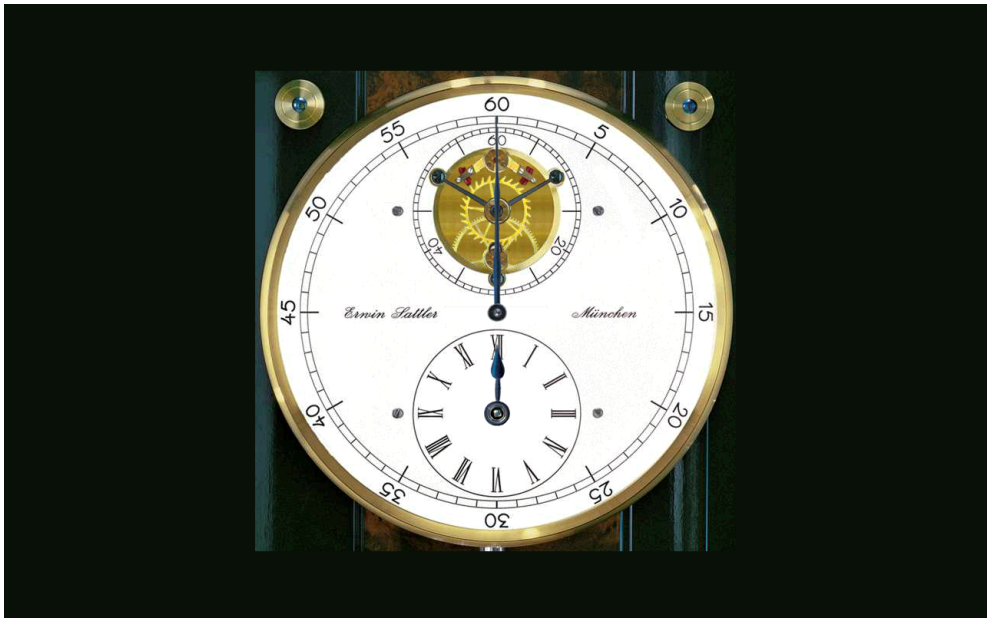


4. Durchsuchen von Graphen

Gesucht sind Prozeduren, die alle Knoten (eventuell auch alle Kanten) mindestens einmal besuchen und möglichst effizient sind.

4.1 Tiefensuche, Depth-First-Search

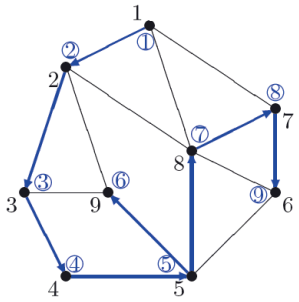
Sei $G = (V, E)$ ein ungerichteter Graph, gegeben als Adjazenzliste.



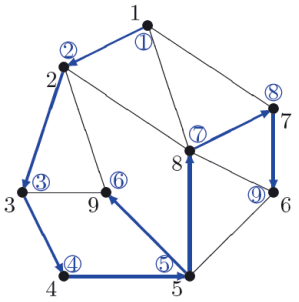
Algorithmus Tiefensuche (DFS):

while $\exists$ unvisited v **do**
  $r :=$ pick (random) unvisited node
  push r onto stack
  while stack $\neq \emptyset$ **do**
    $v :=$ pop top element
    if v unvisited **then**
      mark v visited
      push all neighbours of v onto stack
      perform operations DFS_Ops(v)
    fi
  od
od

Beispiel 299

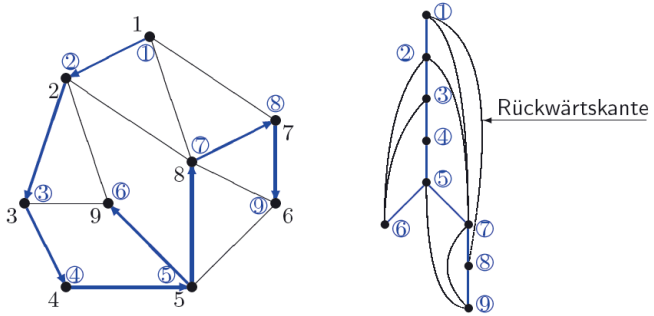


Beispiel 299



Diskrete Strukturen
©Ernst W. Mayr
4.1 Tiefensuche, Depth-First-Search
479/519

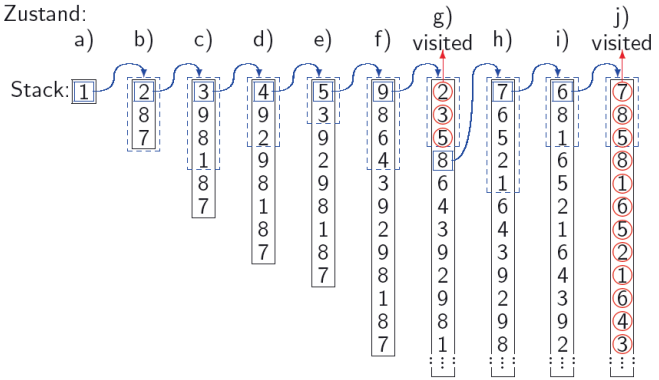
Beobachtung: Die markierten Kanten bilden einen Spannbaum:



Folge der Stackzustände

□ : oberstes Stackelement ⬚ : Nachbarn

○ : schon besuchte Knoten




```
algorithm advanced DFS
  void proc DFSvisit(node v)
    visited[v] := true
    pre[v] := ++precount
    for all u ∈ adjacency_list[v] do
      if not visited[u] then
        type[(v,u)] := 'Baumkante'
        parent[u] := v
        DFSlevel[u] := DFSlevel[v]+1
        DFSvisit(u)
      elsif u ≠ parent[v] then
        type[(v,u)] := 'Rückwärtskante'
      fi
    od
    post[v] := ++postcount
  end proc
```



```
algorithm advanced DFS
  void proc DFSvisit(node v)
    visited[v] := true
    pre[v] := ++precount
    for all u ∈ adjacency_list[v] do
      if not visited[u] then
        type[(v,u)] := 'Baumkante'
        parent[u] := v
        DFSlevel[u] := DFSlevel[v]+1
        DFSvisit(u)
      elsif u ≠ parent[v] then
        type[(v,u)] := 'Rückwärtskante'
      fi
    od
    post[v] := ++postcount
  end proc
```

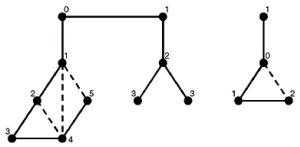
Diskrete Strukturen
©Ernst W. Mayr
4.1 Tiefensuche, Depth-First-Search
483/519

Fortsetzung

```
co Initialisierung: oc
for all v ∈ V do
  visited[v] := false
  pre[v] := post[v] := 0
od
precount := postcount := 0
for all v ∈ V do
  if not visited[v] then
    DFSlevel[v] := 0
    parent[v] := null
    DFSvisit(v)
  fi
od
end
```


Beispiel 300 (gestrichelt sind Rückwärtskanten)

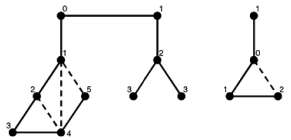
DFS-Level:



Präorder-Nummer:

Beispiel 300 (gestrichelt sind Rückwärtskanten)

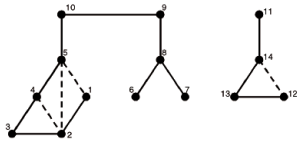
DFS-Level:



Präorder-Nummer:

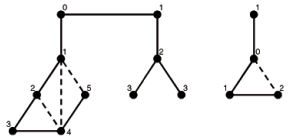
Beispiel (Fortsetzung)

Postorder-Nummer:

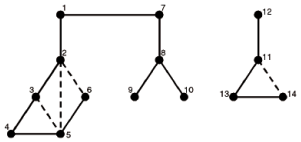


Beispiel 300 (gestrichelt sind Rückwärtskanten)

DFS-Level:

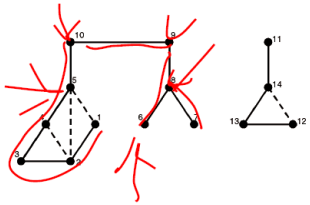


Präorder-Nummer:



Beispiel (Fortsetzung)

Postorder-Nummer:



Beobachtung: Die Tiefensuche konstruiert einen Spannwald des Graphen. Die Anzahl der Bäume entspricht der Anzahl der Zusammenhangskomponenten von G.

Satz 301

Der Zeitbedarf für die Tiefensuche ist (bei Verwendung von Adjazenzlisten)

$$O\big(|V|+|E|\big)\ .$$

Beweis:

Aus Algorithmus ersichtlich. □

Beobachtung: Die Tiefensuche konstruiert einen Spannwald des Graphen. Die Anzahl der Bäume entspricht der Anzahl der Zusammenhangskomponenten von G.

Satz 301

Der Zeitbedarf für die Tiefensuche ist (bei Verwendung von Adjazenzlisten)

$$O\big(|V|+|E|\big)\ .$$

Beweis:

Aus Algorithmus ersichtlich. □

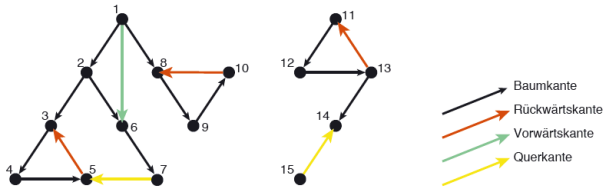
Tiefensuche im Digraphen: Für gerichtete Graphen verwendet man obigen Algorithmus, wobei man die Zeilen

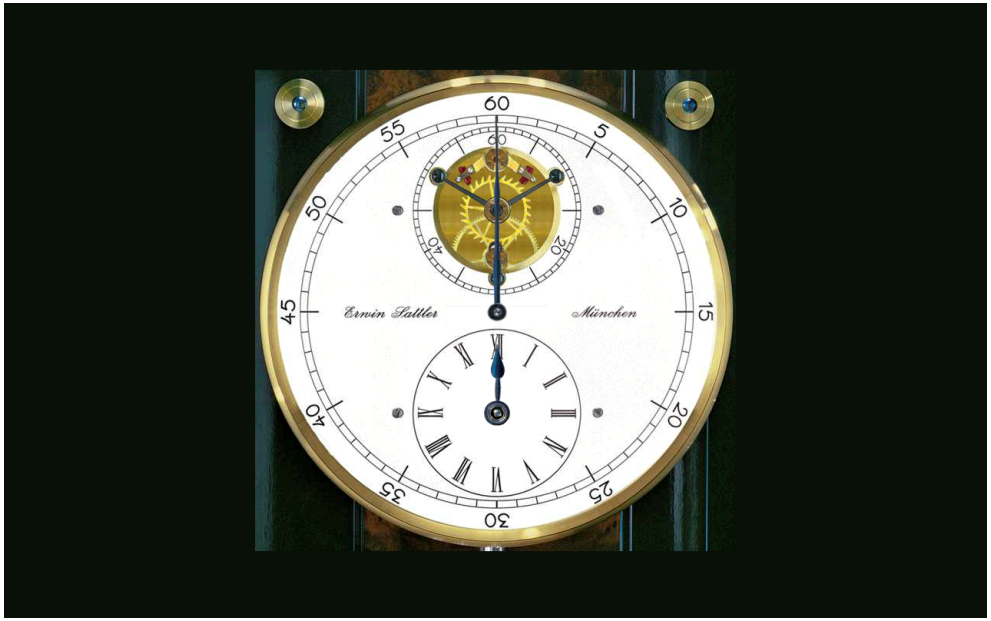
```
elsif u ≠ parent[v] then
  type[(v,u)] := 'Rückwärtskante'
fi
```

ersetzt durch

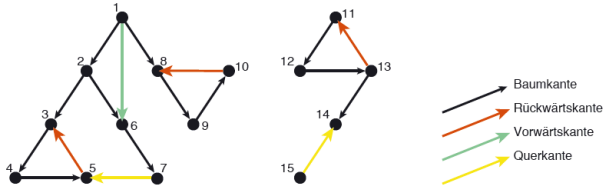
```
elsif pre[u] > pre[v] then
  type[(v,u)] := 'Vorwärtskante'
elsif post[u] ≠ 0 then
  type[(v,u)] := 'Querkante'
else
  type[(v,u)] := 'Rückwärtskante'
fi
```


Beispiel 302 (Präorder-Nummer)



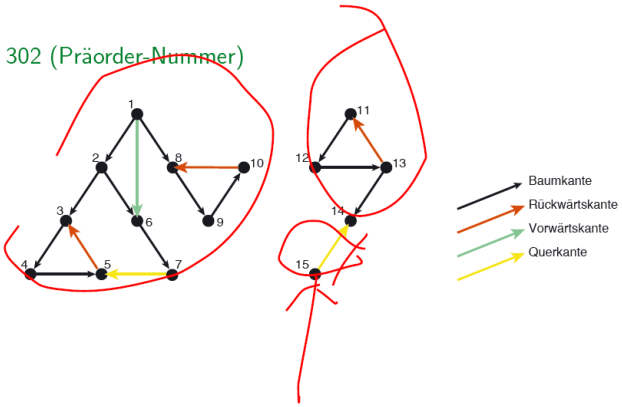


Beispiel 302 (Präorder-Nummer)



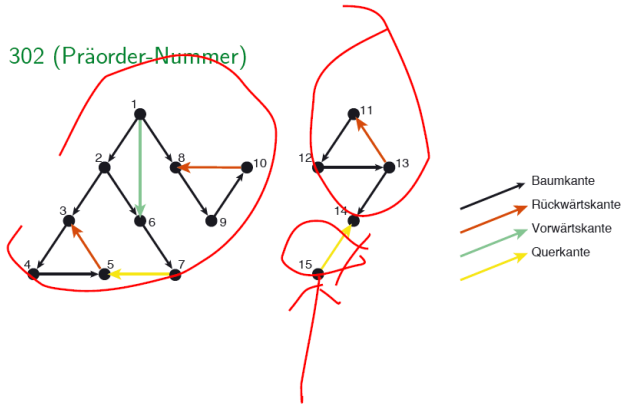
Diskrete Strukturen ©Ernst W. Mayr 4.1 Tiefensuche, Depth-First-Search 489/519

Beispiel 302 (Präorder-Nummer)



Baumkante
Rückwärtskante
Vorwärtskante
Querkante

Beispiel 302 (Präorder-Nummer)



Baumkante
Rückwärtskante
Vorwärtskante
Querkante

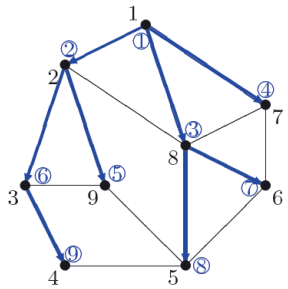
Diskrete Strukturen
©Ernst W. Mayr
4.1 Tiefensuche, Depth-First-Search
489/519

4.2 Breitensuche, Breadth-First-Search

Sei $G = (V, E)$ ein ungerichteter Graph, gegeben mittels Adjazenzlisten.
BFS-Algorithmus:

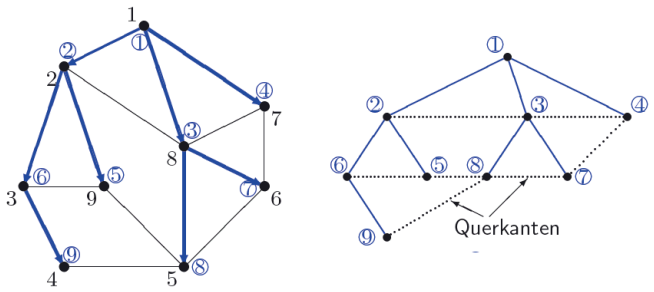
while $\exists$ unvisited v **do**
  $r :=$ pick (random) unvisited node
  push r into (end of) queue
  while queue $\neq \emptyset$ **do**
    $v :=$ remove front element of queue
    if v unvisited **then**
      mark v visited
      append all neighbours of v to end of queue
      perform operations BFS_Ops(v)
    fi
  od
od

Beispiel 303

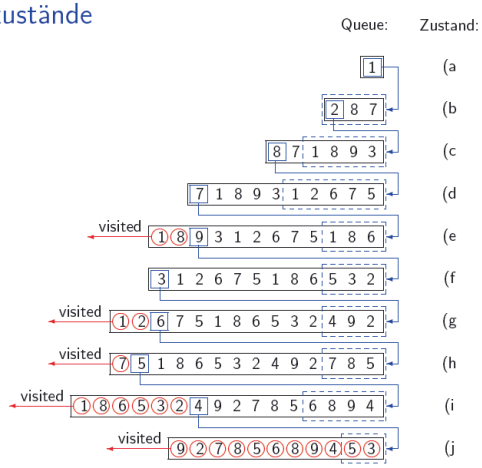


Diskrete Strukturen
©Ernst W. Mayr
4.2 Breitensuche, Breadth-First-Search
491/519

Beobachtung: Die markierten Kanten bilden einen Spannbaum:



Folge der Queuezustände




```
algorithm advanced BFS
  for all v ∈ V do
    touched[v] := false
    bfsNum[v] := 0
  od
  count := 0
  queue := ∅
  for all v ∈ V do
    if not touched[v] then
      bfsLevel[v] := 0
      parent[v] := null
      queue.append(v)
      touched[v] := true
      while not empty(queue) do
        u := remove_first(queue)
        bfsNum[u] := ++count
```


Satz 304
Der Zeitbedarf für die Breitensuche ist (bei Verwendung von Adjazenzlisten)

$$O(|V| + |E|) .$$

Beweis:
Aus Algorithmus ersichtlich. □

Satz 304

Der Zeitbedarf für die Breitensuche ist (bei Verwendung von Adjazenzlisten)

$$O(|V| + |E|) .$$

Beweis:

Aus Algorithmus ersichtlich. □

4.3 Matroide

Definition 305

Sei S eine endliche Menge, $U \subseteq 2^S$ eine Teilmenge der Potenzmenge von S. Dann heißt $M = (S, U)$ ein Matroid und jedes $A \in U$ heißt unabhängige Menge, falls gilt:

1. $\emptyset \in U$
2. $A \in U, B \subseteq A \implies B \in U$
3. $A, B \in U, |B| = |A| + 1 \implies (\exists x \in B \setminus A)\left[(A \cup \{x\}) \in U\right]$

Jede bezüglich $\subseteq$ maximale Menge in U heißt Basis.
Nach 3. haben je zwei Basen gleiche Kardinalität. Diese heißt der Rang $r(M)$ des Matroids.

Beispiel 306

Linear unabhängige Vektoren in einem Vektorraum.

Beispiel 307

G sei folgender Graph:

S = Menge der Kanten von G
U = Menge der kreisfreien Teilmengen von S

Beispiel 306

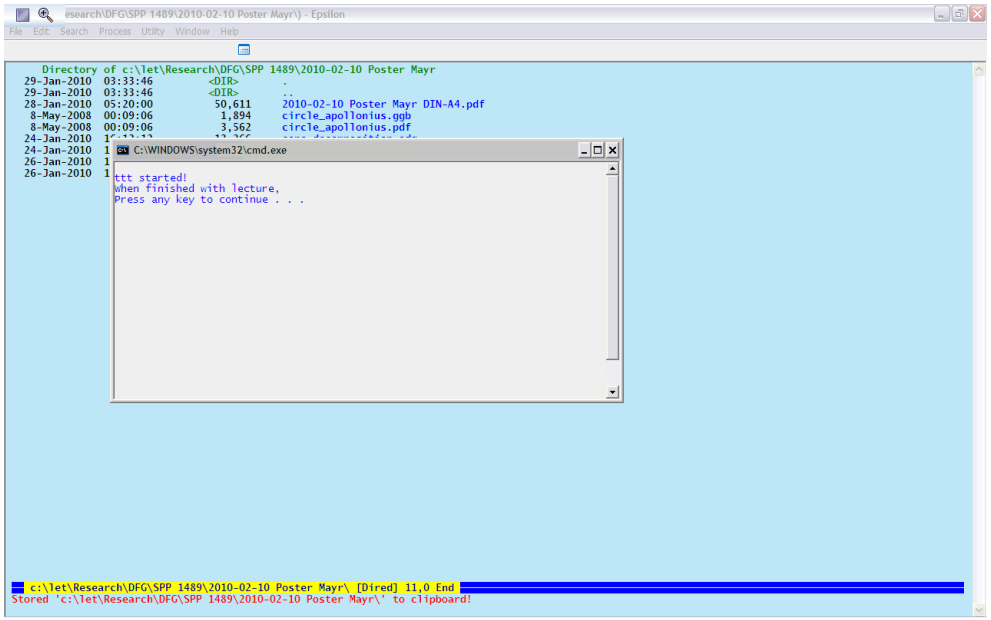
Linear unabhängige Vektoren in einem Vektorraum.

Beispiel 307

G sei folgender Graph:

S = Menge der Kanten von G
U = Menge der kreisfreien Teilmengen von S

Diskrete Strukturen
©Ernst W. Mayr
4.3 Matroide
500/519


esearch\DFG\SPP 1489\2010-02-10 Poster Mayr\) - Epsilon
File Edit Search Process Utility Window Help
Directory of c:\let\Research\DFG\SPP 1489\2010-02-10 Poster Mayr
29-Jan-2010 03:33:46 <DIR> .
29-Jan-2010 03:33:46 <DIR> ..
28-Jan-2010 05:20:00 50,611 2010-02-10 Poster Mayr DIN-A4.pdf
8-May-2008 00:09:06 1,894 circle_apollonius.ggb
8-May-2008 00:09:06 3,562 circle_apollonius.pdf
C:\WINDOWS\system32\cmd.exe
ttt started!
When finished with lecture,
Press any key to continue . . .
c:\let\Research\DFG\SPP 1489\2010-02-10 Poster Mayr\ [Dired] 11,0 End
Stored 'c:\let\Research\DFG\SPP 1489\2010-02-10 Poster Mayr\' to clipboard!