Info2 Uebung vom 24.6.02

Gruppe F, Thomas Briner, http://n.ethz.ch/student/brinerth/info2

24. Juni 2002

1 Nachbesprechung U9

1.1 LAS - Optimalitätsbedingung ?

3	7	13	9	2	3	4	6	5	7	11	1	18	4

Gilt die Optimalitätsbedingung auch für die '7' an Position 9?

3	7	13	9	14	15	4	6	5	7	11	2	12	4

Und jetzt?
→Optimalität muss gelten für die Teilfolge, in der das Element '7' vorkommt.

Analog beim Rucksackproblem:

Gegenstand	Gewicht	Wert
1	3	5
2	7	2
3	16	11
4	28	20
5	30	10

Gewichtslimite ist 57

	0	3	7	10	16	19	23	26	28	30	56	57
1	0	5										
2	0	5	2	7								
3	0	5	2	7	11	16	13	18				
4	0	5	2	7	11	16	13	18	20			
5	0	5	2	7	11	16	13	18	20	10	28	

Optimale Lösung ist also $\{1, 2, 3, 5\}$

Aber: Für die Gewichtslimite 31 ist die Teilmenge der optimalen Lösung für 57 nicht die bestmögliche Lösung

1.2 Implementation für rekursive Lösung

```
(*      Prozedur zum Durchtesten aller möglichen ascending subsequences *)

PROCEDURE naiv(i, last, len: INTEGER; VAR maxLen:INTEGER; a:ARRAY OF INTEGER);
        BEGIN
                IF (i=SIZE-1) THEN
                        IF (len>maxLen) THEN maxLen:=len; END;
                ELSE
                        IF a[i]>=last) THEN
                                naiv(i+1,a[i],len+1,maxLen,a);
                        ELSE
                                naiv(i,last,len,maxLen,a);
                        END;
                END;
        END naiv;
```

1.3 Differenz Huffman - optimaler Suchbaum

Die Aufgabe 9.2 a) hat zwei mögliche Lösungen:

A	B	C	D	E
$\frac{3}{17}$	$\frac{2}{17}$	$\frac{6}{17}$	$\frac{5}{17}$	$\frac{1}{17}$

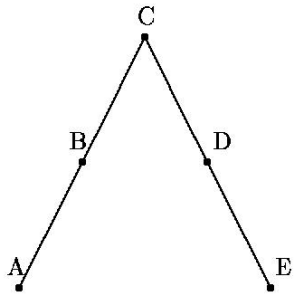
Welche der beiden Lösungen ist besser? →Aufsummieren über die erwartete Codelänge:

- Variante 1: $(\frac{3}{17} * 2) + (\frac{2}{17} * 3) + (\frac{6}{17} * 2) + (\frac{5}{17} * 2) + (\frac{1}{17} * 3) = \frac{37}{17}$
- Variante 2: $(\frac{3}{17} * 3) + (\frac{2}{17} * 4) + (\frac{6}{17} * 1) + (\frac{5}{17} * 2) + (\frac{1}{17} * 4) = \frac{37}{17}$

⇒**beide Lösungen sind gleichwertig**

Der optimale Suchbaum, der entsteht, sieht so aus:

$(-\infty,A)$	A	(A,B)	B	(B,C)	C	(C,D)	D	(D,E)	E	(E,∞)
0	3	0	2	4	6	0	5	0	1	2



2 Vorbesprechung U11

2.1 Median-of-median-Strategie

Problemstellung:

Beispielsweise für Quicksort ist man darauf angewiesen, dass man mit vernünftigem Aufwand ein Element findet, das die ganze Menge in zwei Teile teilt, von denen keiner beliebig klein sein kann.

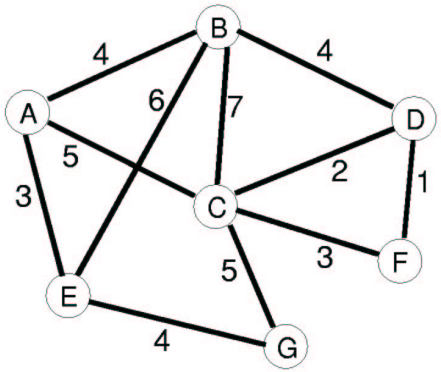
Durch das Median-of-median Verfahren wird sichergestellt, dass immer eine bestimmte Anzahl Elemente grösser und eine bestimmte Anzahl kleiner als das Pivotelement ist.

Vorgehen:

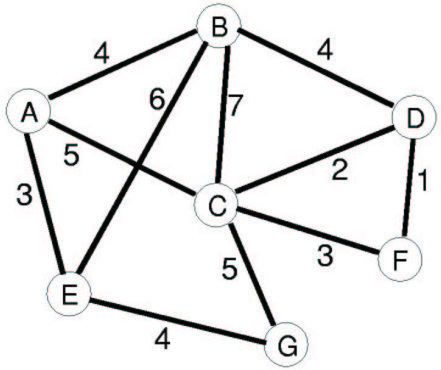
Eine Anwendung dieses Verfahrens ist die Suche nach dem i-ten Element einer unsortierten Menge.

2.2 Minimum Spanning Tree

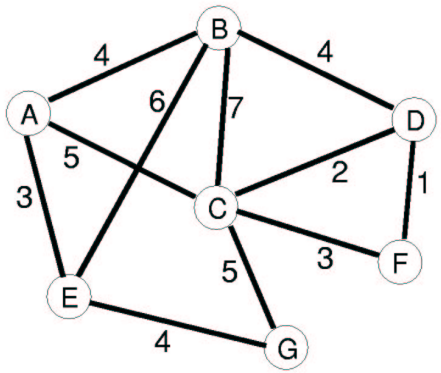
Algorithmus von Kruskal:



Algorithmus von Dijkstra für MST, ausgehend von Knoten A:



Kürzester Weg - Algorithmus von Dijkstra für den Weg von A nach F:



2.3 Implementation von Union-Find

Wie kann festgestellt werden, ob eine Kante einen Kreis bildet?
⇒zu jedem Knoten wird festgehalten, zu welcher Menge (=Zusammenhangskomponente) er gehört

Aus dem Programmskelett:

```
Forest = RECORD
  numNodes : INTEGER;
  prev : POINTER TO ARRAY OF INTEGER;
  size : POINTER TO ARRAY OF INTEGER;
END;
```

Für jeden Knoten wird mit einem Index festgehalten, wer sein 'Vorgänger' ist. Initialisiert wird, indem jeder sein eigener Vorgänger ist.

Bei einem Union(n1,n2) wird zuerst für beide Knoten ein Find(n1), Find(n2) gestartet. Dann werden die beiden Resultate verglichen. Falls sie zur gleichen Menge gehören, kommt das gleiche Resultat raus. Wenn nicht werden sie so zusammengehängt, dass das kanonische Element der kleineren Menge (dasjenige, das bei Find zurückgegeben wird) nun neu nicht mehr auf sich selber zeigt, sondern auf das kanonische Element der grösseren Menge. Es muss also immer die Anzahl Elemente einer Menge mitgeführt werden.

Um zusätzlich zu optimieren, gibt es verschiedene Pfadverkürzungsstrategien beim Find:

- Path collapse
- Path splitting
- Path halfing

2.4 Anmerkung zu 11.3

Mit 'grösserer Baum' bzw. 'kleinerer Baum' ist jeweils die Anzahl der Knoten und nicht die Höhe des Baumes gemeint.