WS 2016 / 2017 Hasam Abazid 3824527

$\boxed{1}$ a)

$$A = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 \end{bmatrix}$$

b) $P = \langle 0, 2, 1, 3, 0 \rangle$
$P = \langle 0, 1, 3, 4, 0 \rangle$
$P = \langle 1, 3, 4, 2, 1 \rangle$
$P = \langle 0, 3, 4, 2, 0 \rangle$
$P = \langle 0, 1, 2, 4, 0 \rangle$

c) $2|E| = 2 \cdot 8 = 16$ Grad.

d) Ein Spannbaum ist eine Kantenteilmenge (T) von der Menge E, sodass der Graph G(V,T) ein Baum ist.

Weg 1

– Gradmatrix –

$$GA = \begin{bmatrix} 4 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 3 \end{bmatrix} - \begin{bmatrix} 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 \end{bmatrix}$$

$$\begin{bmatrix} 4 & -1 & -1 & -1 & -1 \\ -1 & 3 & -1 & -1 & 0 \\ -1 & -1 & 3 & 0 & -1 \\ -1 & -1 & 0 & 3 & -1 \\ -1 & 0 & -1 & -1 & 3 \end{bmatrix} \Rightarrow \begin{vmatrix} 3 & -1 & -1 & 0 \\ -1 & 3 & 0 & -1 \\ -1 & 0 & 3 & -1 \\ 0 & -1 & -1 & 3 \end{vmatrix}$$

$= 45$ Spannbäume

Weg 2: Anzahl der Spannbäume in einem vollständigen Graph ist: n^{n-2}

- und für jede Kante k gibt es die Menge $k = 2n^{n-3}$ Spannbäume

$\Longrightarrow$ Nehmen wir an dass unser Graph vollständig ist $\Longrightarrow$ Spannbäume $= n^{n-2} = 5^3 = 125$

löschen wir eine Kante $\Rightarrow 125 - 2n^{n-3}$

$= 125 - 2(25) = 75$ Spannbäume

Unserem Garph fehlen aber 2 Kanten und hat bestimmt Maximum 70 Spannbäume in diesem fall ?

$\boxed{2}$

a)

1000 Knoten : $2^h - 1$ (Anzahl der Knoten)

$2^9 - 1 = 512 \Longrightarrow 2^9 = 511$ (alle Knoten) (bis 9.Ebene)

$2^{10} - 1 = 1024 \Longrightarrow 2^{10} = 1023$ (alle Knoten) (bis 10.Ebene)

- Der Baum besitzt 10 Ebenen

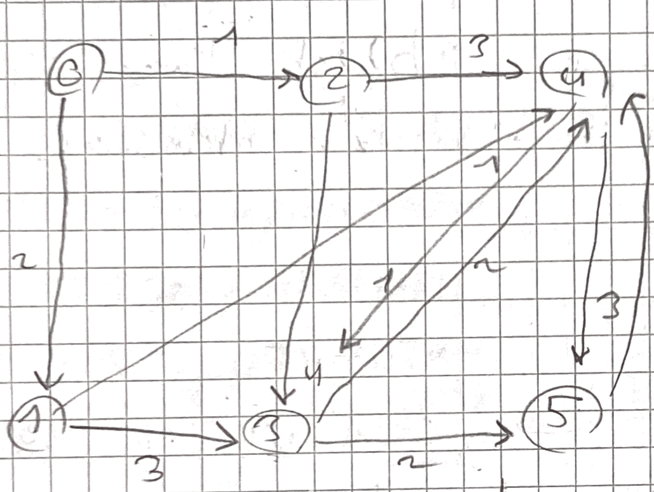
bis die 9-Ebene besitzt der Baum 511 Knoten

Die 10-Ebene hat $1000 - 511 = 489$ Knoten.

b) Der Graph ist ein Binärbaum, also er enthält keine Zyklen. Also jeder Nachfahrer muss eine größere Beginnzeit haben als ihre Vorfahre. Und das kann man im Array Beg erkennen

$\boxed{3}$

a)



b)

dis = [0, 2, 1, ∞, ∞, ∞]
Pre = [-1, 0, 0, -1, -1, -1]
k = <1, 2>

(3) (4)

dis = [0, 2, 1, 5, 4, ∞]
Pre = [-1, 0, 0, 2, 2, -1]
k = <3, 4>

(4)

dis = [0, 2, 1, 5, 3, ∞]
Pre = [-1, 0, 0, 2, 1, -1]
k = <4>

dis = [0, 2, 1, 4, 3, 6]
Pre = [-1, 0, 0, 4, 1, 4]
k = <5, 3>

c)

decreas(Neu schlüssle, wert)

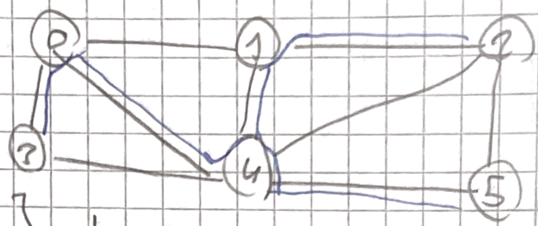
- Reduziert einen schlüssle von einem bereits enthaltenen Wert. Werden die Eigenschaften des MinHeaps verletzt, dann stellt decreas die Eigenschaften wiederher

d) $o(m) = o\left(\frac{n^2-n}{2}\right) \Rightarrow$ im schlechtesten Fall (kompletter Graph)

$= o(n^2)$

$\Rightarrow o(n^2+n) \cdot \log(n)$

$\boxed{4}$



a)
tu = [0, 0, 0, 0, 0]
tv = [3, 0, 0, 0, 0]
r = [0, 1, 2, 0, 4, 5]

tu = [0, 0, 0, 0, 0]
tv = [3, 4, 0, 0, 0]
r = [0, 1, 2, 0, 0, 5]

tu = [0, 0, 4, 0, 0]
tv = [3, 4, 5, 0, 0]
r = [0, 1, 2, 0, 0, 0]

tu = [0, 0, 4, 1, 0]
tv = [3, 4, 5, 4, 0]
r = [0, 0, 2, 0, 0, 0]

tu = [0, 0, 4, 1, 1]
tv = [3, 4, 5, 4, 2]
r = [0, 0, 0, 0, 0, 0]

b) Ich stimme zu, denn ein minimaler Spannbaum ist eine Kantenteilmenge (T) von der Menge (E), sodass der Graph G(T, V) ein Baum ist und W(T) das minimale Gewicht unter allen anderen Spannbäumen.

5

a) Da die Graphen keine Zyklen enthalten können wir für jeweils ein Topologisches Sortieren ausführen, und so bekommen wir ein Array, in dem für alle entstehenden Kanten (u, v) gilt, dass u im Graphen vor v steht und muss zuerst kompiliert werden.

b) Ändern wir ein Datei z.B. "A", dann führen wir eine Breitensuche mit "A" als Startknoten aus. Am Ende der Breitensuche sind die schwarzen gefärbten Knoten die Knoten die übersetzt werden müssen.

c) Wir verwenden den Algorithmus von Kosaraju auf den Gesamten Graphen, um die Starkzusammenhängend Komponenten zu bestimmen