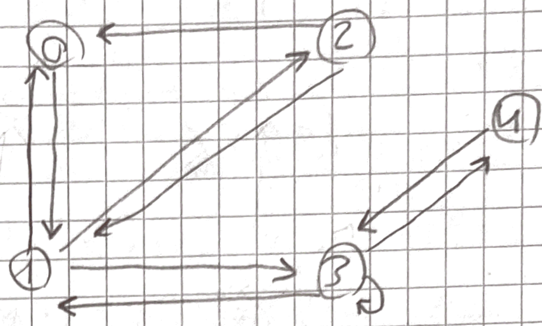
SS 2019 // 2020 Hosam Abozid 3824527

$\boxed{1}$

a)

i)



ii) (0, 1)

[0], [1, 2, 3, 4]

b) - Ausgangsgrad:

Ist die Anzahl der Elemente in der Liste von k

$O(1)$

- Eingangsgrad:

Wir suchen in allen Knoten-listen nach k, im schlimmsten Fall: $O(n^2)$

c)

i) $\frac{n(n-1)}{2} \geq 100$, wir lösen:

$n^2 - n - 200 = 0$

$n = \frac{1 + \sqrt{1 - 4(-200)}}{2} = \frac{1 + \sqrt{801}}{2} \approx 14{,}6$

- ersetzen wir (14):

$\frac{14(13)}{2} = 7 \times 13 = 91$ kanten X

- ersetzen wir (15):

$\frac{15(14)}{2} = 15 \times 7 = 105$ kanten , entfernen wir 5 kanten und bekommen wir 100 kanten

Also 15 knoten

ii) Dann soll der Graph regulär sein:

- für $k=2$

$|V| = \frac{2|E|}{k} \Rightarrow 45 = \frac{200}{2}$

$\Rightarrow 45 = 100$ ↯ (nein)

- für $k=5$

$|V| = \frac{2|E|}{k} \Rightarrow 45 = \frac{200}{5}$

$\Rightarrow 45 = 40$ ↯ (nein)

$\boxed{2}$

a)

col = [1, 0, 0, 0, 0, 0]
dis = [0, ∞, ∞, ∞, ∞, ∞]
pre = [-1, -1, -1, -1, -1, -1]
q = <0>

col = [2, 0, 1, 0, 1, 0]
dis = [0, ∞, 1, ∞, 1, ∞]
pre = [-1, -1, 0, -1, 0, -1]
q = <4, 2>

col = [2, 0, 2, 0, 1, 0]
dis = [0, ∞, 1, ∞, 1, ∞]
pre = [-1, -1, 0, -1, 0, -1]
q = <4>

col = [2, 0, 2, 1, 2, 1]
dis = [0, ∞, 1, 2, 1, 2]
pre = [-1, -1, 0, 4, 0, 4]
q = <5, 3>

col = [2, 1, 2, 2, 2, 1]
dis = [0, 3, 1, 2, 1, 2]
pre = [-1, 3, 0, 4, 0, 4]
q = <1, 5>

col = [2, 1, 2, 2, 2, 2]
dis = [0, 3, 1, 2, 1, 2]
pre = [-1, 3, 0, 4, 0, 4]
q = <1>

col = [2, 2, 2, 2, 2, 2]
dis = [0, 3, 1, 2, 1, 2]
pre = [-1, 3, 0, 4, 0, 4]
q = <>

b) Der Baum hat maximal

- $(2^h - 1)$ knoten
- $(2^h - 2)$ kanten

$\Rightarrow O(2^h - 1 + 2^h - 2) = O(2^{2h} - 3) = \underline{\underline{O(2^{2h})}}$

c) Er muss keine Zyklen enthalten:

- Es sei die kante (u, v) gegeben, ist $beg(u) > beg(v)$, dann ist (u, v) eine Rückwärtskante und wir haben ein Zyklus und der Graph ist nicht Topologisch Sortierbar.

$\boxed{3}$

a)

$$D = \begin{bmatrix} 0 & 4 & \infty & 9 & \infty \\ 3 & 0 & 9 & 8 & 1 \\ 1 & 11 & 0 & 4 & 10 \\ 2 & 7 & \infty & 0 & 8 \\ 8 & 6 & \infty & 6 & 0 \end{bmatrix}$$

$$D^{(0)} = \begin{bmatrix} 0 & 4 & \infty & 9 & \infty \\ 3 & 0 & 9 & 8 & 1 \\ 1 & 5 & 0 & 4 & 10 \\ 2 & 6 & \infty & 0 & 8 \\ 8 & 6 & \infty & 6 & 0 \end{bmatrix}$$

$$D^{(1)} = \begin{bmatrix} 0 & 4 & 13 & 9 & 5 \\ 3 & 0 & 9 & 8 & 1 \\ 1 & 5 & 0 & 4 & 6 \\ 2 & 6 & 15 & 0 & 7 \\ 8 & 6 & 15 & 6 & 0 \end{bmatrix}$$

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

$P_{14} = 2$

$P_{12} = 3$

$P_{13} = 1$

...

c) Er summiert die Einträger für jeden Zeile und wählt den Knoten, deren Zeile das kleinste Gewicht hat. Dieser Knoten ist der Dorf.

$\boxed{4}$ a)

initial	$r = [0, 1, 2, 3, 4, 5, 6, 7]$
nach $u = 0$	$r = [0, 1, 2, 3, 0, 0, 6, 0]$
nach $u = 1$	$r = [0, 1, 2, 1, 0, 0, 6, 0]$
nach $u = 2$	$r = [0, 1, 2, 1, 0, 0, 2, 0]$
nach $u = 3$	$r = [0, 1, 2, 1, 0, 0, 2, 0]$
nach $u = 4$	$r = [0, 1, 2, 1, 0, 0, 2, 0]$
nach $u = 5$	$r = [0, 1, 2, 1, 0, 0, 2, 0]$
nach $u = 6$	$r = [0, 1, 2, 1, 0, 0, 2, 0]$
nach $u = 7$	$r = [0, 1, 2, 1, 0, 0, 2, 0]$

$[0, 4, 5, 7], [1, 3], [2, 6]$

b) Der Algorithmus sucht nach dem r[u] und r[v], und vergleicht r[u] und r[v] und merkt min(r[u], r[v]), und dann sucht im Array nach den werten max(r[u], r[v]) und ersetzt diese Werte durch min(r[u], r[v]).

Im schlechtesten Fall $O(n)$

5.

a)

i) s = ballet , t = blues

	-	b	a	l	l	e	t	
-	0	1	2	3	4	5	6	
b	1	0	1	2	3	4	5	Einfügen von (a) nach (b)
l	2	1	1	1	2	3	4	
u	3	2	2	2	2	3	4	Ersetze (u) durch (l)
e	4	3	3	3	3	2	3	
s	5	4	4	4	4	3	$\boxed{3}$	Ersetze (s) durch (t)

Editdistanz ist 3

ii)
- Einfügen von (a) nach (b)
- Ersetze (u) durch (l)
- Ersetze (s) durch (t).

b) Wir bestimmen die untere- und obere Schranke für jede Zeichenkette mit (S). S = Plus

Pi: $min_1 = 2$, $max_1 = 4$

minus: $min_2 = 1$, $max_2 = 5$

matrix: $min_3 = 2$, $max_3 = 6$

Probability: $min_4 = 7$, $max_4 = 11$

transitivity: $min_5 = 8$, $max_5 = 12$

- Wir merken:

$min_4, min_5 > max_1, max_2, max_3$

$\Rightarrow$ Also, die Zeichenketten "Probability" und "transitivity" können wir ausschießen.

- Wir müssen nun nur für die Zeichenketten "Pi", "minus", "matrix" die Levenshtein-Editierdistanz zu (S) berechnen, da, $min_1, min_2, min_3 < max_1, max_2, max_3$
