

1) Calcule a complexidade de tempo do algoritmo de *Dijkstra*, que retorna o custo do caminho mínimo entre um vértice inicial s e qualquer outro vértice de um grafo. Sendo $G.Adj$ a lista de adjacência que representa as arestas do grafo, w o custo associado as arestas e Q uma lista de vértices. Considere dois cenários:

a) Q é uma lista encadeada.

RELAX (u, v, w)

if $v.d > u.d + w(u, v)$

$v.d \leftarrow u.d + w(u, v)$

$u.pi \leftarrow u$

complexidade da
função RELAX:
 $O(1)$

DIJKSTRA (G, w, s)

for each vertex $v \in G.V$ $O(V)$ repete V vezes

$v.d = \infty$ $O(1)$

$v.pi = \text{NIL}$ $O(1)$

$s.d = 0$ $O(1)$

$S \leftarrow \emptyset$ $O(1)$

$Q \leftarrow G.V$ $O(V)$

while $Q \neq \emptyset$ $O(V)$

$u \leftarrow \text{EXTRACT_MIN}(Q)$ $O(V)$

$S \leftarrow S \cup \{u\}$ $O(1)$

for each vertex $v \in G.Adj[u]$ $O(E)$

RELAX(u, v, w) $O(1)$

todos os vértices de V
são inseridos em Q

→ irá iterar em todos
as arestas "embaladas"
do grupo E

complexidade da função
dijkstra: $O(V^2 + E)$

b) Q é uma lista de prioridade, implementada como um *heap* binário.

Explique a complexidade das linhas em vermelho e dos laços **for** e **while**.

DIJKSTRA (G, w, s)

for each vertex $v \in G.V$ $O(V)$

$v.d = \infty$ $O(1)$

$v.pi = \text{NIL}$ $O(1)$

$s.d = 0$ $O(1)$

$S \leftarrow \emptyset$ $O(1)$

$Q \leftarrow G.V$ $O(V)$

while $Q \neq \emptyset$

$u \leftarrow \text{EXTRACT_MIN}(Q)$ $O(\log V)$

$S \leftarrow S \cup \{u\}$ $O(1)$

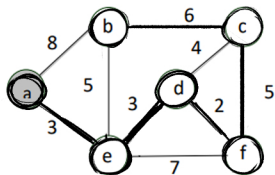
for each vertex $v \in G.Adj[u]$ $O(E)$

RELAX(u, v, w) $O(1)$

A complexidade do algoritmo é
 $O(V \log V + E)$

O extrair min é $\log V$
mas é feito V vezes.

- 2) Mostre cada iteração do algoritmo da questão anterior considerando o grafo abaixo, considere o vértice **a** como o vértice inicial:



6 vértices = 6 iterações

Nome:
Daniella
Vasconcellos

1°
caminho = a

a	b	c	d	e	f
0	8	∞	∞	3	∞

relaxa (e,b), (e,d), (e,f)

2°
caminho: a,e

a	b	c	d	e	f
3	5	∞	3	0	7

relaxa (d,f), (d,c)

3°
caminho: a,e,d

a	b	c	d	e	f
6	3	4	0	3	2

relaxa (c,b)

4°
caminho: a,e,d,f

a	b	c	d	e	f
8	10	5	2	5	0

relaxa (f,c)

5°
caminho: a,e,d,f,c

a	b	c	d	e	f
13	6	0	7	10	11

relaxa (c,b)

6°
caminho: a,e,b,f,c,b

a	b	c	d	e	f
8	0	6	13	5	11

não relaxa arestas

fim do algoritmo.