

- 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.  
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**.

**RELAX** ( $u, v, w$ )

**if**  $v.d > u.d + w(u, v)$   
     $v.d \leftarrow u.d + w(u, v)$   
     $u.pi \leftarrow u$

**DIJKSTRA** ( $G, w, s$ )

**for** each vertex  $v \in G.V$   
     $v.d = \infty$   
     $v.pi = \text{NIL}$   
 $s.d = 0$   
 $S \leftarrow \emptyset$   
 $Q \leftarrow G.V$   
**while**  $Q \neq \emptyset$   
     $u \leftarrow \text{EXTRACT\_MIN}(Q)$   
     $S \leftarrow S \cup \{u\}$   
    **for** each vertex  $v \in G.Adj[u]$   
         $\text{RELAX}(u, v, w)$

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

