Prática 04

Q1 (CLRS 24.3-3)

```
Dijkstra(G, s)
    InitializeSingleSource(G, s)
    Q := new MinQueue(G.V)
    S = {}
    while (! Q.empty()) {
        u := Q.extractMin();
        S := S ∪ {u};
        for each v ∈ G.Adj[u]
            Relax(G.w, u, v)
    }
```

$|Q| > 1$

- O algoritmo ainda estaria correcto?

Invariante do algoritmo de Dijkstra:

- $\forall u \in S.\ d[u] = \delta(s, u)$
  $\wedge\ S = V \setminus Q$

- Seja x o vértice q ficou em Q:
  $\forall u \in V \setminus \{x\}.\ d[u] = \delta(s, u)$

- A operação de relax dos arcos q partem de x não altera o valor de d para nenhum vértice em V.

Q2 (CLRS 24.3-4)

- Verificar se caminhos mais curtos estão bem calculados

- Input: $d[\,]$ e $\pi[\,]$

① Verificar que $d[\,]$ e $\pi[\,]$ estão correctos na origem

- $d[s] = 0 \wedge \pi[s] = Nil$
- $O(1)$

② Verificar q o subgrafo induzido por π é uma árvore

- $G_\pi = (V_\pi, E_\pi)$

$V_\pi = \{ v \mid \pi[v] \neq Nil \} \cup \{ s \}$

$E_\pi = \{ (\pi[v], v) \mid \pi[v] \neq Nil \}$

- DFS_Visit(G_π, s)
  encontrar todos os vértices em V_π e não encontrar nenhum arco para trás

- $O(V+E)$

Q2 (CLRS 24.3-4)

- Verificar se caminhos mais curtos estão bem calculados

- Input: d[ ] e π[ ]

(3) Verificar que d[ ] e π[ ] são consistentes:

  - $\forall v \in V.\ \pi[v] \neq Nil \Rightarrow d[v] = d[\pi[v]] + w(\pi[v], v)$
  - $\forall v \in V \setminus \{s\}.\ \pi[v] = Nil \Leftrightarrow d[v] = \infty$
  - $O(V)$

(4) Verificar q̄ d satisfaz a desigualdade triangular

  - $\forall (u,v) \in E.\ d[v] \leq d[u] + w(u,v)$
  - $\underline{O(V+E)}$

Q2 (CLRS 24.3-4)

· Prova de que o procedimento proposto está correcto

· Suponhamos que existe u tal que $d[u] \neq \delta(s,u)$

Por ① sabemos que $u \neq s$.
Admitimos, sem perda de generalidade, q $d[\pi[u]] = \delta(s, \pi[u])$

· Há dois casos a considerar:

(I) $\pi[u]$ é um predecessor de u num caminho mais curto q liga s a u:

$$\begin{aligned} d[u] &= d[\pi[u]] + w(\pi[u], u) \\ &= \delta(s, \pi[u]) + w(\pi[u], u) \\ &= \delta(s, u) \quad \therefore \end{aligned}$$

Q2 (CLRS 24.3-4)

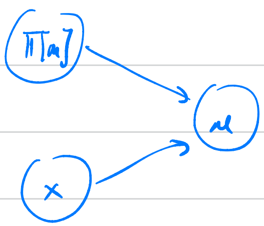
- Prova de que o procedimento proposto está correcto

- Suponhamos que existe u tal que $d[u] \neq \delta(s,u)$

Por ① sabemos que $u \neq s$.
Admitimos, sem perda de generalidade, q $d[\pi[u]] = \delta(s, \pi[u])$

- Há dois casos a considerar:

(II) $\pi[u]$ não é predecessor de u num caminho mais curto que liga s a u. Seja x esse predecessor. Admitimos, sem perda de generalidade, que $d[x]$ está bem calculado.



$$\left.\begin{aligned} d[u] &= d[\pi[u]] + w(\pi[u], u) \\ &= \delta(s, \pi[u]) + w(\pi[u], u) \\ &> \delta(s, u) \\ \\ d[u] &\leq d[x] + w(x, u) \\ &= \delta(s, x) + w(x, u) \\ &= \delta(s, u) \end{aligned}\right\}$$

Q3 (CLRS 24.1-4)

```
BellmanFord'(G, s)
  InitializeSingleSource(G, s)
  for i=1 to |G.V|-1
  |  for each (u,v) ∈ G.E
  |  |  Relax(u, v, G.w)
  let marked = ∅
  for each (u,v) ∈ G.E
  |  if v.d > u.d + w(u,v)
  |  |  marked = marked ∪ {v}
  for each v ∈ marked
    mark all nodes reachable from v with -∞
```

modified DFS

Q4 (CLRS 25.3-3)

Relação entre $\hat{w}$ e w se $w(u,v) \geq 0 \ \forall (u,v) \in E$

- Se $w(u,v) \geq 0 \ \forall (u,v) \in E$, então concluímos que:

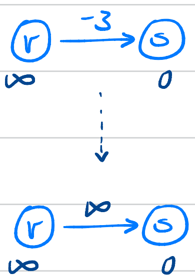
$$\forall v \in V.\ h(v) = 0$$

- Segue que, para todo $(u,v) \in E$:

$$\begin{aligned} \hat{w}(u,v) &= w(u,v) + h(u) - h(v) \\ &= w(u,v) + 0 - 0 \\ &= w(u,v) \end{aligned}$$

Q5 (CLRS 25.3-6)

① No cálculo das alturas de Johnson não precisamos de construir o grafo estendido. Basta escolher com source um vértice s em V.



② O método apresentado em ① funciona se o grafo tiver um único componente.

- Para qualquer $s \in V$, temos que:

$\forall v \in V.\ \delta(s, v) < \infty \Rightarrow$ Alturas bem definidas