

7. Caminhos Mínimos

A15-1

Problema: Dado um grafo com pesos positivos nas arestas e um vértice especial, calcular a distância de cada um dos vértices para o vértice especial.

Seja $G = (V, E)$ um grafo.

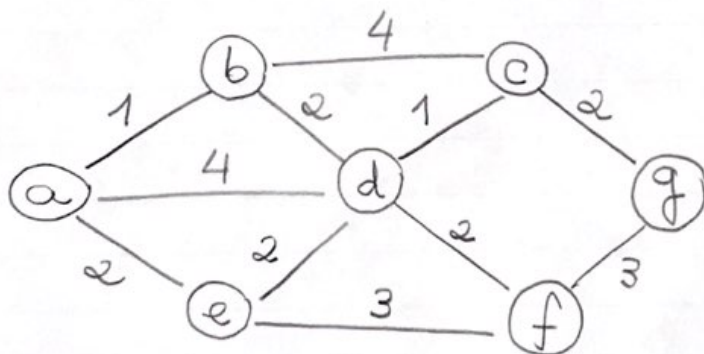
Um caminho P entre dois vértices $u, v \in V$ é uma sequência $u = x_0, x_1, x_2, \dots, x_k = v$ onde dois vértices consecutivos são adjacentes, i.e. $x_i x_{i+1} \in E$.

O comprimento do caminho é $|P| = k$, o número de arestas.

Se G tem pesos nas arestas $w: E \rightarrow \mathbb{Z}$, então

$$|P| = \sum_{0 \leq i < k} w(x_i x_{i+1}).$$

A distância entre u e v é o comprimento do menor caminho entre u e v .



Seja $d(v)$ a distância entre a e v .
 $d(a) = 0$, $d(b) = 1$, $d(e) = 2$, $d(d) = 3$,
 $d(c) = 4$, $d(f) = 5$, $d(g) = 6$

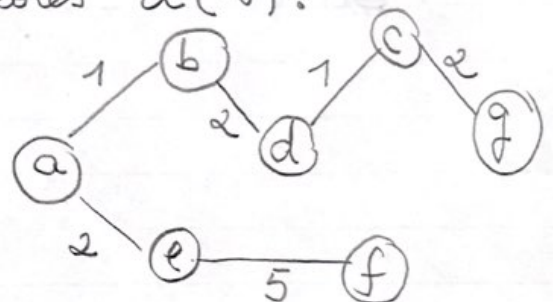
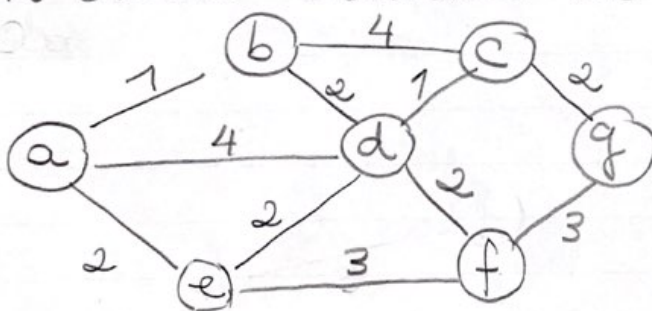
A 15-2

O algoritmo de Dijkstra mantém dois conjuntos de vértices: o conjunto S de vértices explorados e o conjunto $V \setminus S$ de vértices descobertos.

O algoritmo mantém como invariante que:
 o conjunto S contém os vértices para os quais sabemos a distância à origem;
 o conjunto $V \setminus S$ contém para cada vértice a melhor estimativa, o melhor limite superior até o momento.

Para cada $u \in V$, a estimativa $d(u)$ corresponde a caminho que só usa vértices anteriores em S .

O algoritmo agrega vértices v ao conjunto S em ordem crescente de valores $d(v)$.



Produzimos uma árvore geradora de caminhos mais curtos com raiz a .

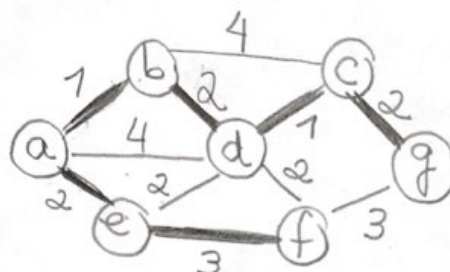
A 15-3

Dijkstra (G, Δ)Para cada vértice v

$$d(v) = \infty$$

Defina conjunto $S = \emptyset$

$$d(\Delta) = 0$$

enquanto $S \neq V$ selecione $u \in V \setminus S$ tal que $d(u)$ é mínimo
adicione u a S para cada vizinho v de u faça
se $d(v) > d(u) + w(uv)$ então
 $d(v) = d(u) + w(uv)$ 

Passo	S	$d(a)$	$d(b)$	$d(c)$	$d(d)$	$d(e)$	$d(f)$	$d(g)$
0	$\emptyset$	0	∞	∞	∞	∞	∞	∞
1	$\{a\}$	—	1	∞	4	2	∞	∞
2	$\{a, b\}$		—	5	3	2	∞	∞
3	$\{a, b, e\}$			5	3	—	5	∞
4	$\{a, b, e, d\}$			4	—		5	∞
5	$\{a, b, e, d, c\}$			—			5	6
6	$\{a, b, e, d, c, f\}$						—	6
7	$\{a, b, e, d, c, f, g\}$							—

ao atualizar $d(v) = d(u) + w(uv)$, guarde a aresta uv para recuperar o caminho mínimo através da árvore de caminhos mais curtos.

Teorema: Considere um vértice $u \in S$.
Em qualquer momento do algoritmo,
 $d(u)$ é a distância entre s e u .

prova: Por indução em $|S|$. A base $|S|=1$
é dada pela inicialização. A hipótese
de indução é assumir válido para $|S|=k \geq 1$.
O passo de indução considera $|S|=k+1$
e o vértice v agregado a S no passo $k+1$.

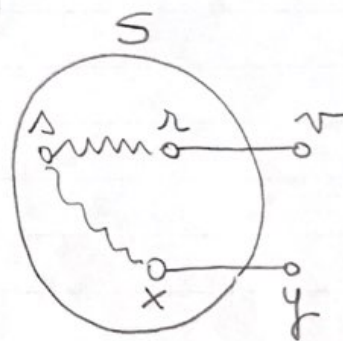
O vértice v foi escolhido em $V \setminus S$ com a
menor $d(v)$ e chamamos de P_v o caminho
de s até v , que satisfaz $|P_v| = d(v)$
e que tem vértices intermediários em S .

Seja $r \in S$ a última aresta de P_v . Quando v
foi agregado a S , temos $r \in S$ e $v \in V \setminus S$.

A hipótese de indução diz que $d(r)$ é a
distância de s a r , e junto com a escolha de
 $r \in S$ e $v \in V \setminus S$, obtemos que P_v é o menor
caminho de s a v que passa por r .

Considere um caminho P de s a v
que não passa por r , e seja xy
a primeira aresta de P
com $x \in S$ e $y \in V \setminus S$.

No passo $k+1$, Dijkstra escolhe v
e não y : $d(v) \leq d(y)$, logo
o subcaminho P' em P de s a y
satisfaz $|P_v| \leq |P'|$,
concluindo $|P_v| \leq |P|$ e a prova. $\square$



A15-5

Prim (G, Δ)

Para cada vértice v

$\text{custo}(v) = \infty$

Defina conjunto $S = \emptyset$

$\text{custo}(\Delta) = 0$

enquanto $S \neq V$

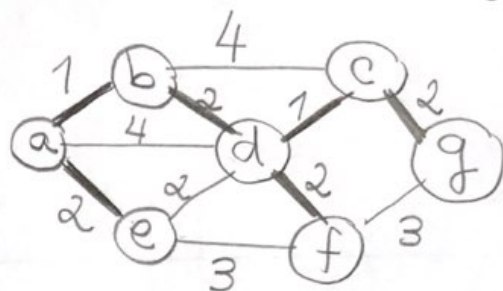
selecione $u \in V - S$ tal que $\text{custo}(u)$ é mínimo

adicione u a S

para cada vizinho v de u faça

se $\text{custo}(v) > w(uv)$ então

$\text{custo}(v) = w(uv)$



Obtemos outra árvore geradora, embora os vértices sejam selecionados na mesma ordem.

A árvore de caminhos mais curtos não é uma MST sempre.

No exemplo, a árvore Prim também é uma árvore de caminhos mais curtos.

Procure exemplo onde: toda árvore de caminhos mais curtos não é MST; toda MST não é árvore de caminhos mais curtos.

Dijkstra é sensível a escolha de s .

O que podemos dizer sobre Prim?