



Algoritmos e Modelação Computacional

Paulo Mateus
MEBiom – LMAC
2018

Árvore da extensão mínima

Considere um grafo pesado $G = (V, E)$ onde cada aresta tem um peso $w : E \rightarrow \mathbb{R}$.

Problema: Encontrar um subgrafo acíclico T que contém todos os vértices de G tal que o seu peso

$$w(T) = \sum_{e \in T} w(e)$$

é mínimo.

Nota: Este subgrafo acíclico forma uma árvore que se chama árvore de extensão mínima (AEM).

AEM

Um **corte** de um grafo G é uma partição $\{S, V \setminus S\}$ de V .

Uma aresta **cruza** um corte se a sua origem e destino se encontram em partes diferentes da partição.

Uma aresta diz-se **leve** para um corte $\{S, V \setminus S\}$ se o seu peso é mínimo entre todas as arestas que cruzam o corte.

Teorema 7 *Seja $S \subseteq T$ onde T é uma AEM e seja e uma aresta leve para $\{S, V \setminus S\}$, então existe uma AEM T' tal que $S \cup \{e\} = S' \subseteq T'$.*

Algoritmo de Prim

Entrada nó inicial u

1 A cada nó v associa-se

a) $C[v]$ o custo mínimo para v (inicializado a $+\infty$)

b) $E[v]$ a aresta que com esse custo (inic. a null)

2) F inicializa-se como uma floresta com nó u e $C[u]=0$

3) Q os vértices não incluídos em F (inic com $V \setminus \{u\}$)

4) while(!Q.vaziaQ()){

a) encontrar a aresta leve $e=\{x,v\}$ para o corte $\{F,Q\}$ com x em F

b) $Q.remove(v)$; // $F.add(v)$;

c) $E[v]=e$; $C[v]=g.weight(e)$ //não é nec para proj

d) $F.addedge(e)$

5) return F

