



ARBORESCÊNCIAS

Marcus Ritt

INF 5016/CMP 588 – Algoritmos avançados — Ago 2021 1

1. Revisão

2. Dicas

- Podemos **subtrair uma constante** do custo de todos arcos entrantes de um vértice.
- Sempre vamos manter que o arco entrante de **menor custo é 0** (em particular depois de contrações).

1. No grafo atual: **seleciona** um arco entrante de peso 0 de cada vértice, exceto a raiz.
2. Caso este grafo não possui **ciclo**: adiciona todos arcos ao grafo suporte H , e inicia fase II.
3. Senão: **identifica um ciclo**, e adiciona os arcos do ciclo ao grafo suporte H .
4. **Contraí o ciclo** (aplicando a transformação de pesos).
5. Resolve o problema recursivamente.

- Nas chamadas recursivas não podemos copiar e contrair o grafo.
- Vamos facilitar a implementação por **destruir** o grafo de entrada (i.e. contrair vértices mesmo)

Para **contrair** um ciclo C numa representação com listas de adjacência:

1. Seleciona um vértice representante $\bar{r} \in C$.
2. Para todos demais vértices $\bar{v} \in C$ do ciclo:
 1. Para arcos saíntes (\bar{v}, \bar{w}) : adiciona um arco (\bar{r}, \bar{v}) , caso $\bar{w} \notin C$
 2. Para arcos entrantes (\bar{u}, \bar{v}) : adiciona um arco (\bar{u}, \bar{r}) , caso $\bar{u} \notin C$
 3. Remove vértice \bar{v} do grafo.
3. Para o "novo" vértice \bar{r} : transforma os pesos dos arcos entrantes.

Também é necessário, para todo arco \bar{a} manter o **arco original** a do grafo de entrada.

- A chamada recursiva vai retornar uma arborescência no grafo contraído (nota que depois da chamada recursiva o grafo original foi destruído)
- Essa arborescência tem que "traduzida" para os arcos originais antes da chamada recursiva.
- Depois podemos adicionar os arcos do ciclo, exceto o único arco do ciclo que entre no vértice que já tem arco entrante