

Lista de soluções 2

Questão 1

Considere duas tarefas subsequentes i e j na solução. Para um tempo de término C da tarefa j eles contribuem um custo

$$w_i(C - p_j) + w_j C$$

para um função. Caso trocamos a ordem para ji , obtemos um custo

$$w_j(C - p_i) + w_i C.$$

A ordem ij é melhor que ji caso

$$\begin{aligned} w_i(C - p_j) + w_j C &< w_j(C - p_i) + w_i C \\ \iff w_i/p_i &\geq w_j/p_j. \end{aligned}$$

Logo a solução ótima pode ser obtida por (gulosamente) selecionar a tarefa de maior w_j/p_j em tempo $O(n \log n)$.

Questão 2

No que segue vamos supor que todas arestas possuem um peso diferente; essa hipótese pode ser eliminada de acordo com Kleinberg e Tardos [1, p. 149]. Caso o grafo possua mais que uma componente conexa, a solução é dada pela floresta de árvores geradoras mínimas (AGMs) dos componentes (um argumento simples por contradição). Caso o grafo é conexo, podemos fazer duas observações gerais.

1. A solução ótimo possui exatamente dois componentes (caso contrário podemos remover uma aresta).
2. Os dois componentes da solução ótima formam AGMs (senão podemos encontrar uma solução menor).

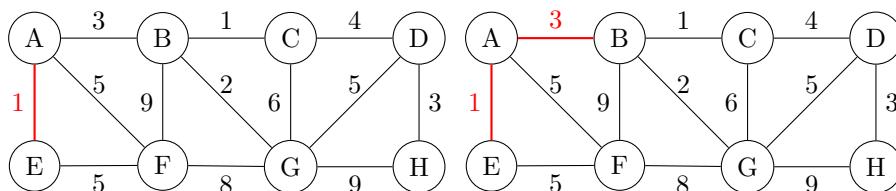
Vamos provar que uma solução ótima S poder ser obtido removendo a aresta mais pesada a de uma AGM T (i.e. $S = T \setminus \{a\}$). Sejam S_1 e S_2 os componentes de S , e supõe temos uma solução S' com componentes S'_1 e S'_2 de custo $c(S') < c(S)$. Logo

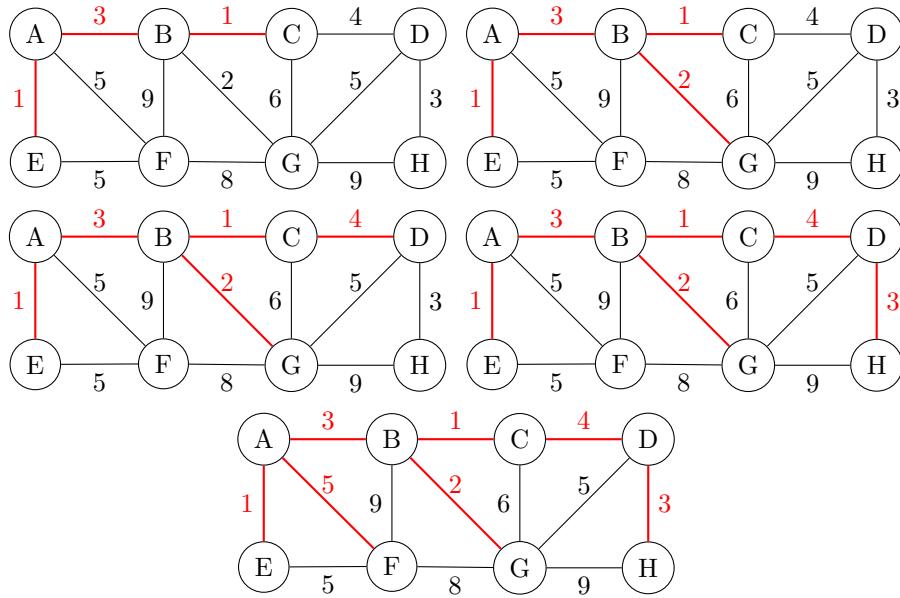
1. A aresta a não pode conectar S'_1 e S'_2 , senão a árvore geradora $S' \cup \{a\}$ teria custo $c(S' \cup \{a\}) = c(S') + c_a < c(S) + c_a = c(T)$, uma contradição com a minimalidade de T .
2. Sem perda de generalidade, supõe que a conecta dois vértices de S'_1 . Como a conecta S_1 e S_2 , $S'_1 \cap S_1 \neq \emptyset$ e $S'_1 \cap S_2 \neq \emptyset$, e como S'_1 é conexo, $S'_1 \cup \{a\}$ possui um ciclo. Logo existe uma aresta a' em S'_1 que conecta S_1 e S_2 . Pela minimalidade de S'_1 , $p_{a'} < p_a$. Mas então encontramos uma aresta a' mais leve que a , e $c(S \cup \{a'\}) = c(S) + c_{a'} < c(S) + c_a = c(T)$, novamente uma contradição com a minimalidade de T .

A complexidade do algoritmo é claramente é igual com a complexidade de qualquer algoritmo que determina uma AGM.

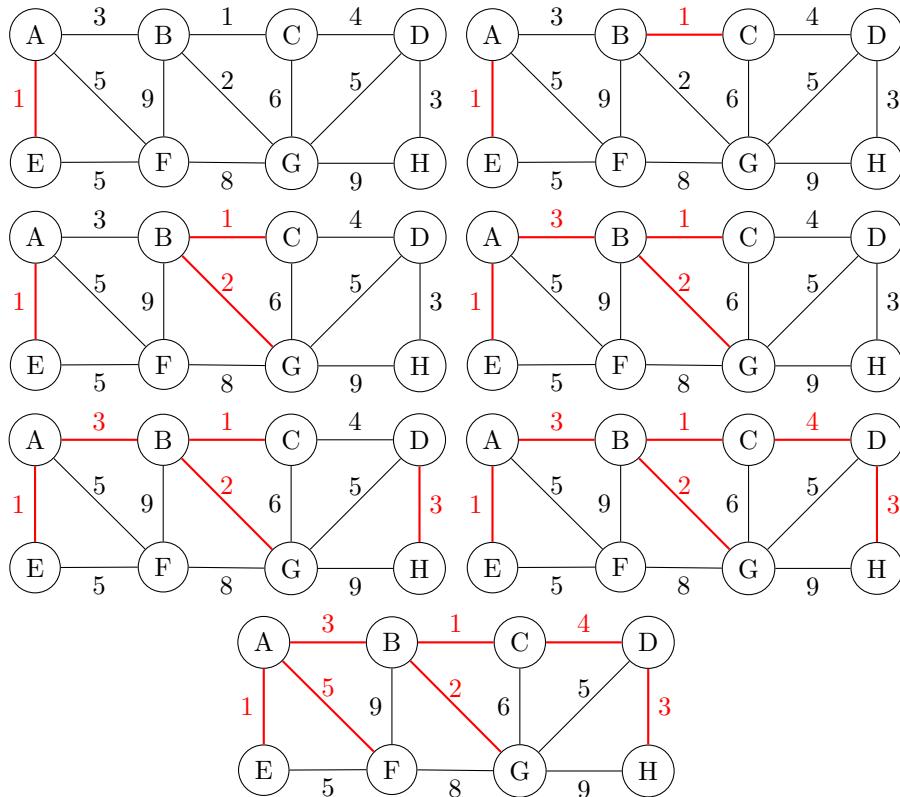
Questão 3

- A sequência de arestas selecionadas pelo algoritmo de Prim a partir do vértice A é





A sequência de arestas escolhidas pelo algoritmo de Kruskal é



Questão 4

- Vamos aplicar o teorema de Akra-Bazzi com $a_1 = a_2 = 1$, $b_1 = \epsilon$, $b_2 = \epsilon$, $h_1(x) = \lfloor \epsilon x \rfloor - \epsilon x$, $h_2(x) = \lceil (1 - \epsilon)x \rceil - (1 - \epsilon)x$, e $g(x) = cx$. As condições do teorema são satisfeitos, porque $|h_1(x)| \leq 1$, $|h_2(x)| \leq 1$ e $|g'(x)| = c \in O(1)$. Logo obtemos $p = 1$ por resolver $\epsilon^p + (1 - \epsilon)^p = 1$ e

temos

$$T(x) = \Theta\left(x + \left(1 + \int_1^x c/u du\right)\right) = \Theta(x \log x).$$

2. Vamos aplicar o teorema de Akra-Bazzi com $a_1 = k - k'$, $a_2 = k'$, $b_1 = 1/k$, $b_2 = 1/k$, $h_1(x) = \lfloor k/n \rfloor - k/n$, $h_2(x) = \lceil k/n \rceil - k/n$, e $g(x) = ckn$. As condições do teorema são satisfeitos, porque $|h_1(x)| \leq 1$, $|h_2(x)| \leq 1$, e $|g'(x)| = kc \in O(1)$, para k constante. Logo obtemos $p = 1$ por resolver $(k - k')/k^p + k'/k^p = k/k^p = 1$ e temos

$$T(x) = \Theta\left(x + \left(1 + \int_1^x ck/u du\right)\right) = \Theta(x \log x).$$