MAC0338 - Entrega da lista 6/7

Algoritmos gulosos - Exercício 2

A entrada é uma sequência de números $x_1, x_2, \ldots, x_n$ onde n é par. Projete um algoritmo que particione a entrada em $n/2$ pares da seguinte maneira. Para cada par, computamos a soma de seus números. Denote por $s_1, s_2, \ldots, s_{n/2}$ as $n/2$ somas. O algoritmo deve encontrar uma partição que minimize o máximo das somas e deve ser $\mathrm{O}(n \log n)$.

Resposta:

Algorithm 1: MinimizaSomas(x[n])

Function *MINIMIZA-SOMAS(x[n])*:
- ordena(x) ; ▷ `O(n log n)`
- cria vetor vazio s com índices de 1 até $n/2$; ▷ `O(n)`
- para $i \leftarrow 1$ até inclusive $n/2$: ; ▷ `O(n/2)`
    - $s_i \leftarrow x[i] + x[n-i+1]$; ▷ `O(1)`
- **return** s ; ▷ `O(1)`

Corretude

Vamos provar a corretude do algoritmo acima.
Ordene $x_1 \leq \cdots \leq x_n$.
Caso (x_1, x_n) forme um par, então já temos o que queremos.
Caso contrário, suponha que (a_1, a_k) seja um par tal que $r \neq n$, e (a_n, a_l) seja um par tal que $s \neq 1$. Para fazer a escolha gulosa, troque a_k com a_n e a_l com a_1. Agora temos (a_1, a_n) e (a_k, a_l). Note que

- dentre os pares antigos (a_1, a_k) e (a_n, a_l), $a_1 + a_k \leq a_n + a_l$;
- a soma do novo par $a_1 + a_n$ é menor ou igual a $a_n + a_l$, já que $a_1 \leq a_l$;
- a soma do novo par $a_k + a_l$ é menor ou igual a $a_n + a_l$, já que $a_k \leq a_n$.

Portanto, a escolha gulosa adotada não aumenta o máximo de nenhuma soma.
Agora, considere a sequência sem x_1 nem x_n, ou seja, $x_2, \ldots, x_{n-1}$. Usando o mesmo raciocínio, os pares sempre serão da forma (x_i, x_{n-i+1}) e a soma máxima irá se manter mínima.

Consumo de tempo

Note que, de acordo com as anotações no algoritmo 1, o consumo de tempo é

$$O(n \log n) + O(n) + (O(n/2) \cdot O(1)) + O(1) = O(n \log n),$$

como queríamos.

MST - Exercício 1

(CRLS Ex. 23.1-1) Seja e uma aresta de custo mínimo em um grafo G com custos nas arestas. É verdade que e pertence a alguma MST de G? É verdade que e pertence a toda MST de G? Sua justificativa **não** pode ser baseada nos algoritmos de Kruskal ou Prim.

Resposta:

Sim, é verdade que e pertence a alguma MST de G.

Suponha que e não pertença a uma MST T. Adicionando e em T, teremos um ciclo. Remova uma aresta $f \neq e$ que também está no ciclo e teremos uma nova MST T'. Por definição de e, f tem custo estritamente maior que e, então o custo de T é estritamente maior que o custo de T'. Achamos uma MST com custo menor que T! Mas como T era de custo mínimo por definição, logo chegamos a uma contradição. Portanto, e deve pertencer a uma MST de G.

Não, não é verdade que e pertence a todas as MSTs de G.

Eis um contraexemplo. Suponha que um grafo G cíclico com todas as arestas de pesos iguais. Logo, todas as arestas têm peso mínimo, mas como G tem um ciclo, logo todas as MSTs de G **não** terão todas as arestas. Logo, não é verdade que e pertence a todas as MSTs de G.

MST - Exercício 4

(CRLS Ex. 23.1-2) Prove ou desprove a seguinte afirmação: Dado um grafo G com pesos nas arestas, um conjunto de arestas A de G, e um corte que respeita A, toda aresta que cruza o corte e que é segura para A tem peso mínimo dentre todas as arestas desse corte.

Resposta:

A afirmação é falsa. Eis um contraexemplo.