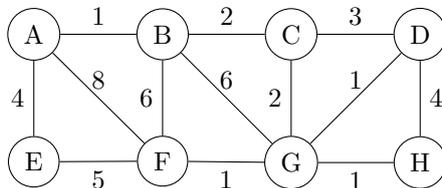


Lista de exercícios algoritmos gulosos

Entrega: 9/4/2011

Exercício 1

Suponha que desejamos encontrar a árvore espalhada mínima do seguinte grafo.



- Execute o algoritmo de Prim; sempre que houver uma escolha de nós, use a ordem alfabética (por exemplo comece pelo nó A). Desenhe uma tabela mostrando os valores intermediários dos custos (distâncias) dos nós.
- Execute o algoritmo de Kruskal no mesmo grafo. Mostre como fica a coleção de conjuntos em cada passo do algoritmo.

Exercício 2

É dado um grafo $G = (V_G, A_G)$ com pesos de aresta positivos e uma árvore geradora mínima $T = (V_T, A_T)$ com relação a estes pesos; você pode considerar que G e T são dados como listas de adjacência. Agora suponha que o peso de uma particular aresta $a \in A$ seja modificado de p_e para um novo valor p'_e . Você quer rapidamente atualizar a árvore geradora mínima T para refletir a mudança, sem recomputar a árvore inteira do nada. Existem quatro casos. Em cada caso forneça um algoritmo de tempo linear para atualizar a árvore.

- $e \notin A_T$ e $p'_e > p_e$.
- $e \notin A_T$ e $p'_e < p_e$.
- $e \in A_T$ e $p'_e < p_e$.
- $e \in A_T$ e $p'_e > p_e$.

Exercício 3

Considere o seguinte problema de sequenciamento em uma máquina: são dados n tarefas com tempo de execução p_i para $i \in [n]$. Cada tarefa tem que ser executada sem interrupção, e em cada instante a máquina só pode executar uma tarefa. Além disso, cada tarefa possui um *deadline* d_i . É desejável que toda tarefa termine antes do seu *deadline*, mas isso nem sempre é possível. Caso a execução da tarefa i termine em C_i (e logo iniciou em $C_i - p_i$) o seu *atraso* é $L_i = \max\{C_i - d_i, 0\}$. Logo, caso execução da tarefa termine antes do *deadline*, o atraso é 0; caso contrário o atraso é o parte do tempo de execução depois do *deadline*. Queremos encontrar uma sequência em que a máquina pode executar todas tarefas que minimiza o maior atraso $L_{\max} = \max_{i \in [n]} L_i$.

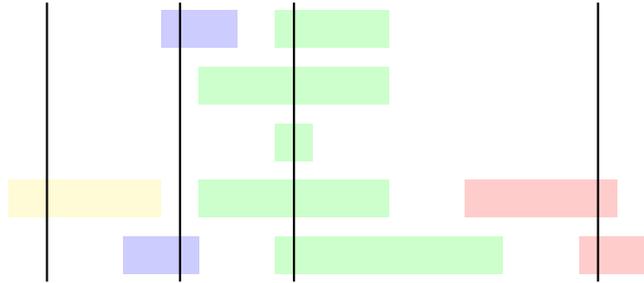
- Encontra a sequência ótima e determina o seu valor L_{\max} para a seguinte instância.

Tarefa	1	2	3	4
p_i	5	4	3	6
d_i	3	5	11	12

- Forneça um algoritmo guloso que encontra a sequência ótima em tempo $O(n \log n)$.

Exercício 4

É dado um conjunto de intervalos de tempo $S = \{(c_i, f_i) \mid i \in [n]\}$. Vamos dizer que um conjunto de pontos P *pica* S caso cada intervalo em S contém pelo menos um ponto em P . Queremos encontrar um conjunto que pica S com o menor número de pontos possíveis. Forneça um algoritmo que encontra tal conjunto em tempo polinomial. Segue um exemplo com menor conjunto de tamanho 4.



Exercício 5 (Quebra-cabeça, opcional)

É dado uma matriz com $n \times m$ números reais. Você pode em cada passo multiplicar todos números de uma coluna ou linha da matriz por -1 . Mostre que existe uma sequência dessas operações tal que no final a soma dos números em cada linha e em cada coluna é positiva. Exemplo:

$$\begin{pmatrix} -1 & 3 & 2 \\ -\pi & -2 & 1 \\ 1 & 1 & 1 \end{pmatrix} \rightarrow \begin{pmatrix} 1 & 3 & 2 \\ \pi & -2 & 1 \\ -1 & 1 & 1 \end{pmatrix}.$$

Lembrança:

- Para receber pontos as afirmações tem que ser provadas, em particular a corretude dos algoritmos propostos.
- Os exercícios podem ser resolvidos em colaboração com outros, mas a entrega é individual informando os eventuais colaboradores.
- A entrega é eletrônica, não escrito a mão, em formato PDF.
- Somente entregam respostas que vocês sabem explicar pessoalmente.