Departamento de Ciência de Computadores FCUP

Desenho e Análise de Algoritmos (CC211) 2012/13

2º Teste Escrito (12.12.2012) *duração: 3h*

N.º ______ Nome ______

1. Utilizando os símbolos $\subseteq$ ou $\subset$, relacione as classes $\Theta(n^2 \log n)$, $O(n^4)$, $\Omega(n^2)$ e $O(1)$, se comparáveis. Justifique formalmente a resposta, recorrendo às definições das notações O, Θ e Ω.

2. Considere uma estrutura de dados semelhante à dada nas aulas para implementação de uma heap de mínimo (com informação adicional). A estrutura tem três campos x, y, e n, sendo x e y *arrays* e n um inteiro que indicará o número de elementos que estão na heap. Cada elemento de x tem um par de valores $p : id$ em que p define a prioridade desse elemento e id um identificador que lhe está associado. O elemento que está na posição v de y tem o índice da posição que v ocupa na heap. Se $y[v] = 0$ então v não está na heap. Nem $y[0]$ nem $x[0]$ serão usadas.

a) Apresente em pseudocódigo um algoritmo para alterar a prioridade de um elemento v para *novap*, sabendo que *novap* é menor do que o seu valor atual. Deve ter complexidade temporal $O(\log_2 n)$ e preservar as propriedades de uma heap de mínimo e da estrutura definida. Use, por exemplo, $x[k].p$ e $x[k].id$, $y[v]$ e n para referir variáveis, omitindo a identificação da estrutura correspondente.

b) Suponha que tem doze elementos numerados de 1 a 12, e que o estado da estrutura é o seguinte com $n = 11$.

x :	–	3 : 10	4 : 11	7 : 5	10 : 3	7 : 4	10 : 7	15 : 2	20 : 9	24 : 1	8 : 6	10 : 8	

y :	–	9	7	4	5	3	10	6	11	8	1	2	0

Desenhe a árvore correspondente e justifique que se trata de uma heap de mínimo. Desenhe também a árvore que resulta da aplicação do algoritmo que apresentou na alínea anterior se $v = 6$ e $novap = 3$, e assinale na árvore o(s) elemento(s) testado(s) ou alterado(s) na transformação.

c) Use propriedades das heaps para mostrar que a complexidade do algoritmo é $O(\log_2 n)$.

3. Considere uma variante do problema "Caixotes de morangos", com m caixas para distribuir e n lojas, conhecendo o lucro do envio de c caixas para a loja i, para $1 \leq i \leq n$ e $1 \leq c \leq m$ (com lucros inteiros positivos). Há que distribuir os caixotes pelas lojas de forma a maximizar o lucro.

```
t[0] ← 0;
ler_vec(y, m);        // ler y[1]...y[m]
Para cada k ← 1 até m fazer
    t[k] ← y[k];
Para cada i ← 2 até n fazer
    ler_vec(y,m);     // loja i
    j ← m;
    Enquanto j ≥ 1 fazer
        Para cada k ← 0 até j − 1 fazer
            Se t[j] < t[k] + y[j − k] então
                t[j] ← t[k] + y[j − k];
        j ← j − 1;
```

a) Apresente a recorrência $t(j, i)$ que define o lucro máximo que se pode obter quando se distribui j caixas pelas primeiras i lojas, com para $1 \leq i \leq n$ e $0 \leq j \leq m$. Explique.

b) O algoritmo dado à esquerda calcula o valor de $t(m, n)$, quando os lucros são dados por lojas. Justifique a correção do algoritmo, começando por definir com rigor o estado das variáveis t e y no final de cada iteração do ciclo "Enquanto" e do segundo ciclo "Para". **NB: Não é necessário uma prova por indução.**

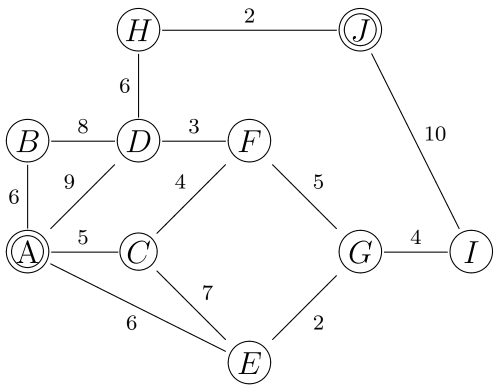
c) (*perg. alternativa a 8.)) Caracterize a complexidade do algoritmo em função de m e n. Explique os detalhes.

4. Que problema resolve o algoritmo de Prim em grafos não dirigidos com pesos nos ramos? O que é um grafo conexo? De que modo o algoritmo de Prim pode ser usado para verificar se o grafo dado é conexo? Que vantagem ou desvantagem tem face ao algoritmo de pesquisa em largura? E, ao algoritmo de pesquisa em profundidade? E, ao algoritmo de Kosajaru (se se começar por transformar o grafo num grafo dirigido simétrico)?

5. Justifique que qualquer caminho mínimo de x para y num grafo com distâncias positivas é ou um arco (x, y) ou é constituído por caminhos mínimos. Explique de que modo esta propriedade é explorada na determinação dos caminhos mínimos nos algoritmos de Dijkstra (de um nó para todos), Floyd-Warshall (de todos para todos) e Bellman-Ford (de todos para todos).

6. Dê um exemplo de uma rede de fluxo (com seis nós) para um problema de fluxo máximo (de um nó s para t). Represente um fluxo f na rede que não seja óptimo e tal que o algoritmo de Edmonds-Karp obteria o fluxo óptimo em <u>**duas**</u> iterações a partir de f. Explique (e ilustre).

7. Considere uma rede representada por um grafo não dirigido $G = (V, E, p, \{s, t\})$ com valores associados aos ramos, sendo $p(e)$ um inteiro positivo, qualquer que seja $e \in E$, por exemplo, como na figura, e s (origem) e t (destino) dois vértices especiais.



Suponha que G é de facto uma rede de transportes e $p(e)$ o valor da temperatura mínima (esperada) no troço e. Pretende-se um algoritmo para encontrar um caminho óptimo de s para t, sendo *óptimo* qualquer caminho tal que a temperatura mínima registada ao longo do caminho é máxima quando considerados todos os caminhos alternativos de s para t (na figura $s = A$ e $t = J$).

a) Que relação tem este problema com os abordados nas aulas?

b) Apresente um algoritmo $O((n + m) \log_2 n)$ para resolver o problema sendo $m = |E|$ e $n = |V|$.

c) Que estruturas de dados utilizou para garantir essa complexidade? Que propriedades explora para garantir a correção do algoritmo?

d) Determine a solução para o exemplo usando o algoritmo.

8. (*<u>perg. alternativa a 3.c)</u>) Suponha que tem um grafo dirigido acíclico $G = (V, E, p)$ com pesos e em que os <u>identificadores dos nós já definem uma ordem topológica</u>. Suponha que um grupo de pessoas vai ser transportado de s (origem) para t (destino) e que será dividido por vários veículos. Cada veículo assegura uma *ligação* (arco da rede) e apenas uma. Todos os veículos são usados apenas uma vez, e os veículos que saem de um nó sairão exatamente ao mesmo tempo (se necessário, aguardam a chegada dos que demoram mais tempo a chegar). Admitindo que $p(u, v)$ designa o tempo que demorará a efectuar o troço (u, v) e que o transbordo de pessoas nos nós é muito rápido (podendo ser negligenciado) pretende-se determinar ao fim de quantas horas o grupo estará novamente reunido em t.

a) Apresente um algoritmo que resolva o problema em $\Theta(m + n)$, sendo $n = |V|$ e $m = |E|$.

b) Altere o grafo que está no exemplo dado no problema anterior, substituindo cada ramo por um arco que deverá ser orientado de u para v se a letra que identifica u preceder a que identifica v no alfabeto. Aplique o algoritmo a esse exemplo para $s = A$ e $t = J$. (FIM)