

Exame de Algoritmos e Estruturas de Dados II

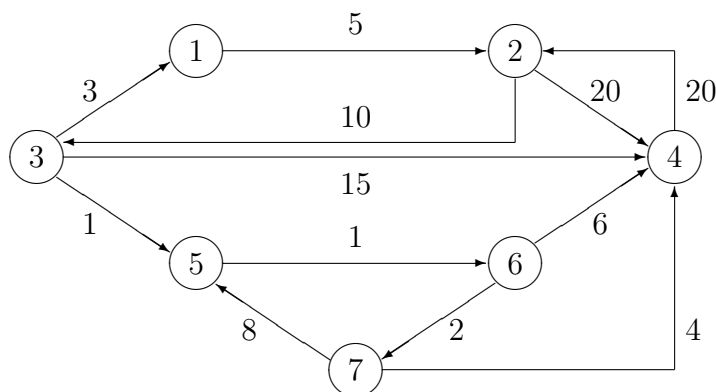
Departamento de Informática

Universidade Nova de Lisboa

25 de Janeiro de 2007

Por favor, entregue as respostas em folhas distintas.

1. Considere uma árvore B^+ de ordem 4 e capacidade 4, com 1 024 elementos.
 - (a) [1.5 valores] Calcule o número mínimo de chaves que podem estar nos nós internos da árvore. Justifique a sua resposta.
 - (b) [2.5 valores] Calcule o número mínimo e o número máximo de páginas (folhas ou nós internos) que a árvore pode ter. Justifique a sua resposta, indicando a altura da árvore, em ambos os casos.
2. [4 valores] Seja G um grafo orientado e pesado, cujos arcos têm custo positivo. Dados dois vértices, v e w , pretende-se seleccionar um caminho entre v e w que tenha o menor comprimento pesado e que, de entre os caminhos que satisfazem a restrição anterior, passe pelo menor número de arcos.



Por exemplo, no grafo representado na figura, há dois caminhos de menor comprimento pesado entre os vértices 3 e 4.

Comprimento Pesado	Caminho
8	3 5 6 4
8	3 5 6 7 4

Neste caso, o caminho seleccionado teria que ser 3 5 6 4, que passa por 3 arcos, porque o caminho 3 5 6 7 4 percorre mais arcos.

Apresente um algoritmo (em pseudo-código) que, dados um grafo $G = (V, A)$ orientado e pesado, cujos arcos têm custo positivo, e dois vértices, v e w , determine um caminho entre v e w nas condições referidas acima. Para simplificar, pode assumir que há, pelo menos, um caminho entre v e w . Estude a complexidade temporal do seu algoritmo, no pior caso.

3. [3 valores] Relembre o tipo abstracto de dados **Partição** e as duas implementações dadas, denominadas *união por altura* e *união por dimensão*. Apresente a representação gráfica de duas árvores, A e B , que satisfaçam simultaneamente as quatro seguintes propriedades.

- As árvores A e B podem ser obtidas através de uma sequência de operações de união por altura (a partir de uma partição acabada de criar).
- As árvores A e B podem ser obtidas através de uma sequência de operações de união por dimensão (a partir de uma partição acabada de criar).
- Ao efectuar uma união por altura entre as árvores A e B , a raiz da nova árvore é a raiz de A (independentemente da ordem dos argumentos na chamada da operação de união).
- Ao efectuar uma união por dimensão entre as árvores A e B , a raiz da nova árvore é a raiz de B (independentemente da ordem dos argumentos na chamada da operação de união).

4. [3 valores] Seja $G = (V, A)$ um grafo não orientado. Para qualquer vértice $x \in V$, o *grau* de x é o número de arcos que possuem uma extremidade em x :

$$\text{grau}(x) = \#\{(u, v) \in A \mid x = u \text{ ou } x = v\}.$$

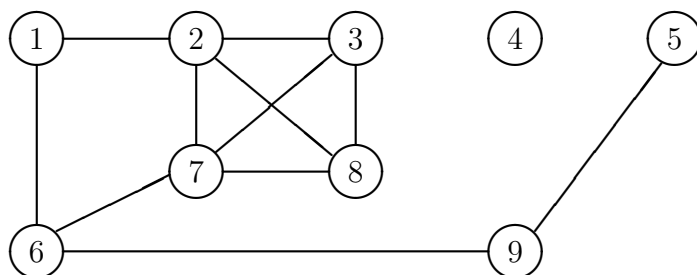
O grafo diz-se *cúbico* se, e só se, qualquer vértice tiver grau 0 ou grau 3:

$$(\forall x \in V) \text{ grau}(x) = 0 \text{ ou } \text{grau}(x) = 3.$$

Por exemplo, no grafo esquematizado na figura, o grau de 1 é 2, o grau de 2 é 4 e o grau de 4 é 0. Portanto, o grafo não é cúbico. Repare, no entanto, que o subgrafo

$$(\{1, 2, 3, 4, 5, 6, 7, 8, 9\}, \{(2, 3), (2, 7), (2, 8), (3, 7), (3, 8), (7, 8)\})$$

é um grafo cúbico.



O **Problema do Subgrafo Cúbico** formula-se da seguinte forma.

Dado um grafo não orientado $G = (V, A)$, existe um subconjunto não vazio de arcos $A' \subseteq A$ tal que $G' = (V, A')$ é um grafo cúbico?

Prove que o Problema do Subgrafo Cúbico é **NP**.

5. [3 valores] Seja X uma sequência não vazia de números inteiros. Denotaremos por $\text{Pos}(X)$ a soma dos elementos positivos de X e por $\text{Neg}(X)$ a soma dos elementos negativos de X . Por convenção, $\text{Pos}(X) = 0$ se não existirem inteiros positivos em X e $\text{Neg}(X) = 0$ se não existirem inteiros negativos em X .

Por exemplo, se X for $\langle 3\ 3\ 0\ 1 \rangle$, $\text{Pos}(X) = 7$ e $\text{Neg}(X) = 0$.

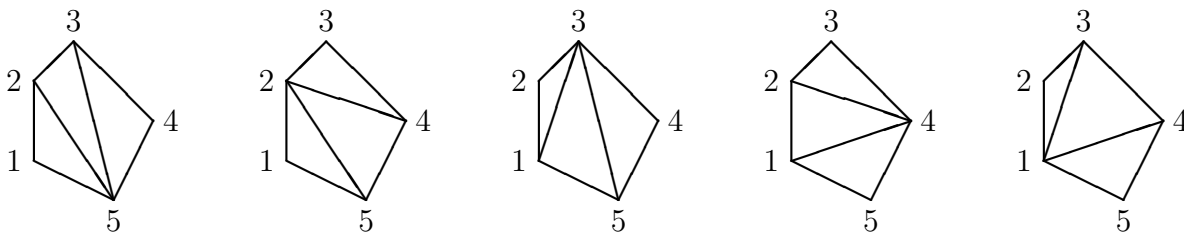
Considere a função recursiva $F_X(i, s)$, onde $X = \langle x_1\ x_2\ \dots\ x_n \rangle$ é uma sequência não vazia de números inteiros, i é um inteiro entre 1 e n , e s é um inteiro entre $\text{Neg}(X)$ e $\text{Pos}(X)$.

$$F_X(i, s) = \begin{cases} 0, & \text{se } i = 1 \text{ e } s \neq x_1; \\ 1, & \text{se } i = 1 \text{ e } s = x_1; \\ \max(F_X(i-1, s-x_i-1), \\ \quad F_X(i-1, s-x_i), \\ \quad F_X(i-1, s-x_i+1)), & \text{se } i > 1 \text{ e } \text{Neg}(X) < s-x_i < \text{Pos}(X); \\ F_X(i-1, s), & \text{nos restantes casos.} \end{cases}$$

Apresente um algoritmo, desenhado segundo a técnica da programação dinâmica, que, dados um inteiro positivo N , uma sequência $X = \langle x_1\ x_2\ \dots\ x_N \rangle$ de inteiros, e um inteiro S entre $\text{Neg}(X)$ e $\text{Pos}(X)$, calcula o valor da função $F_X(N, S)$. Estude a complexidade temporal e espacial do seu algoritmo, no pior caso.

6. [3 valores] Há várias formas de *partir* um polígono convexo em triângulos, traçando um conjunto de segmentos de recta, chamados *diagonais*. Cada diagonal liga dois vértices não consecutivos do polígono. Em cada partição do polígono, as diagonais só se podem intersectar num vértice.

Por exemplo, para partir um pentágono basta traçar duas diagonais, havendo cinco formas distintas de o fazer, como se mostra na figura.



O *custo de uma partição* do polígono é a soma dos comprimentos das respectivas diagonais. Uma partição cujo custo seja o menor possível diz-se uma *partição de custo mínimo*.

Apresente **uma função recursiva** que, dado um polígono convexo, definido pela sequência das coordenadas dos seus vértices, ordenada no sentido dos ponteiros do relógio,

$$P = \langle (x_1, y_1), (x_2, y_2), \dots, (x_n, y_n) \rangle \quad (\text{com } n \geq 4),$$

calcula o custo das partições de P de custo mínimo. Pode recorrer à função que calcula a distância entre dois pontos, $d((x, y), (x', y'))$, sem a definir. Indique claramente o que representa cada uma das variáveis que utilizar e explicita a chamada inicial.

Sugestão: Repare que cada lado do polígono e, em particular, o lado que liga (x_1, y_1) a (x_n, y_n) , pertence a um único triângulo.