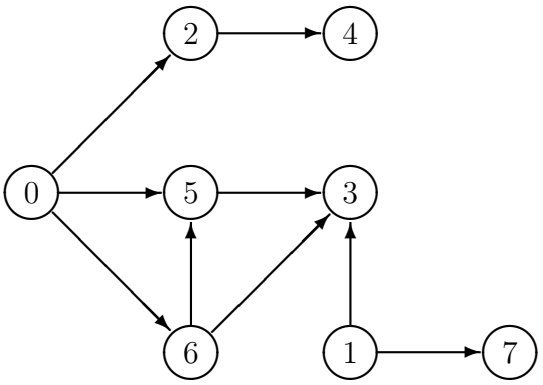
1º Teste de Análise e Desenho de Algoritmos

Departamento de Informática

Universidade Nova de Lisboa

13 de Abril de 2013

1. Suponha que se executa a **versão iterativa** do algoritmo que percorre em profundidade os vértices de um grafo, com o grafo G esquematizado na figura.



Assuma que os métodos *vertices* e *outAdjacentVertices* iteram sempre os vértices por ordem crescente. Por exemplo, $G.outAdjacentVertices(0)$ produz os vértices 2, 5 e 6 (por esta ordem). Indique:

(a) [2 valores] a ordem pela qual os vértices são percorridos (passados a *TREAT* como argumento);

(b) [1 valor] o número de vezes que o método *dfsExplore* é executado; e

(c) [2 valores] para cada vértice do grafo, o número de vezes que esse vértice é empilhado.

2. O vector *partition* contém o estado de uma partição (i.e., o estado de uma instância da classe *UnionFindInArray*). Assuma que a partição tem duas árvores, cujas raízes são 0 e 5.

partition:	?	0	0	0	0	?	5	5	7
	0	1	2	3	4	5	6	7	8

(a) Assuma que se adoptou **união por dimensão** na implementação da partição.

- [1 valor] Quais são os valores de *partition*[0] e *partition*[5]?
- [1 valor] Indique o estado da partição (ou seja, o conteúdo do vector *partition*) após a execução do método *union*(0, 5).

(b) Assuma que se adoptou **união por altura** na implementação da partição.

- [1 valor] Quais são os valores de *partition*[0] e *partition*[5]?
- [1 valor] Indique o estado da partição após a execução do método *union*(0, 5).

3. [6 valores] Considere a seguinte função recursiva $f(i,j)$, onde i e j são números inteiros positivos.

$$f(i,j) = \begin{cases} 1 & \text{se } i = 1 \text{ e } j \geq 1; \\ i\, f(i-1,j) - 1, & \text{se } i \geq 2 \text{ e } j = 1; \\ \min\limits_{1 \leq k < \min(i,j)} \Big(f(k,j) + f(i-k, j-k) \Big), & \text{se } i \geq 2 \text{ e } j \geq 2. \end{cases}$$

Apresente um algoritmo, desenhado segundo a técnica da programação dinâmica, que, dados dois números inteiros positivos m e n, calcula o valor de $f(m,n)$. Estude (justificando) as complexidades temporal e espacial do seu algoritmo, no pior caso.

4. [5 valores] Na *República do Faz-de-Conta* (RFC), a unidade monetária é o *toste*. Troca-se dinheiro estrangeiro por tostes nos bancos da RFC, mas não é possível trocar tostes por outra moeda, nem é permitido sair do país com tostes. Por este motivo, os estrangeiros gastam sempre os últimos tostes nas lojas de *souvenirs* do aeroporto. No entanto, como já despacharam a bagagem de porão, a grande maioria pretende adquirir o menor número possível de *souvenirs*.

Para facilitar o processo de escolha dos *souvenirs*, todas as lojas acordaram no conjunto dos preços que é possível atribuir a um *souvenir*. A tabela com este conjunto de preços é divulgada publicamente.

Por exemplo, se os preços acordados fossem os que se encontram na Tabela 1, seria possível gastar 1 752 tostes adquirindo apenas cinco *souvenirs* (um de 1 000 tostes, dois de 375 tostes e dois de 1 toste).

1	5	10	50	100	200	375	500	1 000	2 000	3 500	5 000	7 500	10 000

Tabela 1: Tabela dos preços possíveis dos *souvenirs*.

Apresente **uma função recursiva** que, com base:

- num inteiro positivo V e
- numa sequência $P = (p_1, p_2, \ldots, p_n)$ crescente e não vazia de inteiros distintos, onde $p_1 = 1$,

calcula o número mínimo de *souvenirs* que é necessário adquirir para gastar os V tostes, quando os preços possíveis dos *souvenirs* são $(p_1, p_2, \ldots, p_n)$. Indique claramente o que representa cada uma das variáveis que utilizar e explicite a chamada inicial.