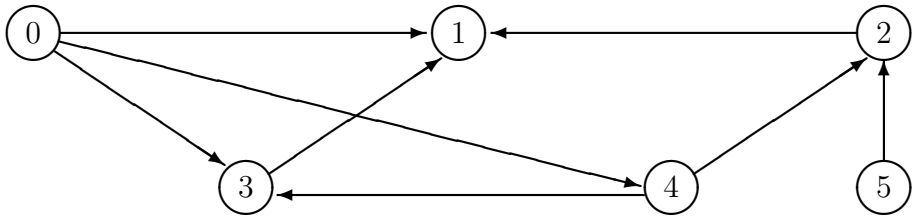
Recurso de Análise e Desenho de Algoritmos

Departamento de Informática
Universidade Nova de Lisboa
11 de Julho de 2009

1. [3 valores] Suponha que se executa a **versão iterativa** do algoritmo que percorre os vértices de um grafo em profundidade, com o grafo G esquematizado na figura.

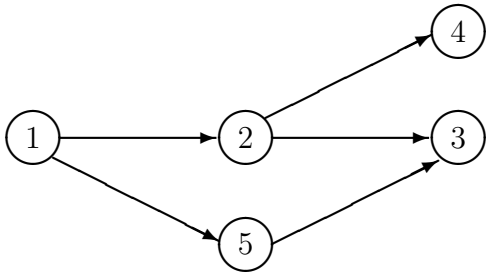


   Assuma que os métodos *vertices* e *outAdjacentVertices* iteram sempre os vértices por ordem crescente. Por exemplo, $G.outAdjacentVertices(0)$ produz os vértices 1, 3 e 4 (por esta ordem). Indique:

   - a ordem pela qual os vértices são percorridos (passados a *TREAT* como argumento);
   - o número de vezes que o método *dfsExplore* é executado; e
   - para cada vértice do grafo, o número de vezes que esse vértice é empilhado.

2. [4 valores] Considere um grafo orientado e acíclico, onde cada vértice representa uma tarefa de um projecto. A existência de um arco do vértice v para o vértice w indica que a tarefa v terá de estar concluída antes de se iniciar a tarefa w.

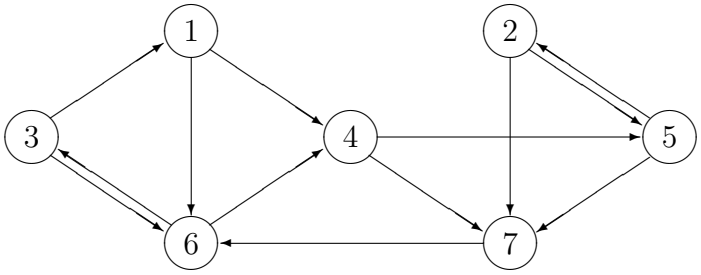
   Como se quer adiar o mais possível uma dada tarefa T, pretende-se saber qual é o maior número de tarefas que podem ser realizadas antes de se executar T. Por exemplo, no caso do grafo desenhado na figura, antes de se executar a tarefa 5, podem ser realizadas, no máximo, três tarefas: as tarefas 1, 2 e 4.



   Escreva um algoritmo (em pseudo-código) que, dados um grafo orientado e acíclico, com a informação sobre as tarefas do projecto, e uma dessas tarefas, calcula o número máximo de tarefas que podem ser realizadas antes de se executar a tarefa dada. Estude (justificando) a complexidade temporal do seu algoritmo, no pior caso.

3. [4 valores] Seja G um grafo orientado e fracamente conexo. Um *circuito Hamiltoniano* em G é um circuito simples que passa por todos os vértices de G.

   Por exemplo, 1 4 5 2 7 6 3 1 é um circuito Hamiltoniano no grafo esquematizado na figura.



   O **Problema do Circuito Hamiltoniano Orientado** formula-se da seguinte forma. Dado um grafo G, orientado e fracamente conexo, existe um circuito Hamiltoniano em G?

   Prove que o Problema do Circuito Hamiltoniano Orientado é NP-completo.

4. [3 valores] Considere a função recursiva $F_{X,Y}(i,j)$, onde:

   - $X = (x_1\, x_2\, \cdots\, x_m)$ e $Y = (y_1\, y_2\, \cdots\, y_n)$ são duas sequências não vazias de inteiros;
   - i e j são inteiros não negativos tais que $i \leq m$ e $j \leq n$; e
   - $G(a,b)$ é uma função definida para números inteiros que vale a, quando $a \leq b$, e vale $a - b$, no caso contrário.

$$F_{X,Y}(i,j) = \begin{cases} i, & \text{se } i \geq 0 \text{ e } j = 0; \\ j, & \text{se } i = 0 \text{ e } j > 0; \\ \min(\ F_{X,Y}(i-1,j-1) + G(x_i,y_j),\quad F_{X,Y}(i-1,j) + 2, & \\ \qquad F_{X,Y}(i,j-1) + 3\ ), & \text{se } i > 0 \text{ e } j > 0. \end{cases}$$

   Apresente um algoritmo, desenhado segundo a técnica da programação dinâmica, que, dadas duas sequências não vazias de números inteiros,

$$X = (x_1\, x_2\, \cdots\, x_m) \quad \text{e} \quad Y = (y_1\, y_2\, \cdots\, y_n),$$

   calcula o valor da função $F_{X,Y}(m,n)$. Estude (justificando) a complexidade temporal e espacial do seu algoritmo, no melhor caso, no pior caso e no caso esperado.

(Continue, porque o exame tem mais duas perguntas.)

5. [3 valores] Considere a classe *QueueInTwoStacks*, das filas com disciplina FIFO, de elementos do tipo E, implementadas com duas pilhas.

```
public class QueueInTwoStacks<E>
{
    private Stack<E> in;
    private Stack<E> out;

    public QueueInTwoStacks( )
    {
        in = new StackInList<E>();
        out = new StackInList<E>();
    }

    public boolean isEmpty( )
    {
        return in.isEmpty() && out.isEmpty();
    }

    public void enqueue( E element )
    {
        in.push(element);
    }

    public E dequeue( ) throws EmptyQueueException
    {
        if ( this.isEmpty() )
            throw new EmptyQueueException();

        if ( out.isEmpty() )
            while ( !in.isEmpty() )
                out.push( in.pop() );

        return out.pop();
    }
}
```

Considere também a função $\Phi(F)$, que atribui a cada fila F da classe *QueueInTwoStacks* o número de elementos guardados na pilha F.in:

$$\Phi(F) = \text{size}(F.\text{in})\,.$$

Prove que Φ é uma função potencial válida e calcule a complexidade amortizada dos métodos *isEmpty*, *enqueue* e *dequeue*, justificando. No estudo da complexidade amortizada do método *dequeue*, assuma que a fila não está vazia, mas analise separadamente os casos em que a pilha *out* está vazia e não está vazia. Assuma que todos os métodos efectuados sobre as pilhas (criação, *isEmpty*, *push* e *pop*) são constantes.

6. [3 valores] Numa pequena cidade do interior, o jornal local é distribuído ao domicílio, por uma empresa que dispõe de C colaboradores. O tempo (em minutos) que um colaborador demora a fazer a distribuição dos jornais depende do número de jornais que distribui e varia de colaborador para colaborador. O objectivo da empresa é minimizar o tempo total da distribuição de J centenas de jornais (ou seja, pretende-se encurtar, tanto quanto possível, o tempo de entrega "do último jornal"). Para simplificar, admita que a unidade (indivisível) de jornais é a centena.

   Como exemplo (c.f. a tabela abaixo), assuma que existem 3 colaboradores ($C = 3$) e que o número total de jornais a distribuir é 500 ($J = 5$). Caso a empresa decidisse entregar 1 centena de jornais ao Colaborador 1, 3 centenas de jornais ao Colaborador 2 e 1 centena de jornais ao Colaborador 3, o tempo total da distribuição dos 500 jornais seria de 20 minutos.

Centenas de Jornais	**Colaborador 1**	**Colaborador 2**	**Colaborador 3**
1	6	10	20
2	21	16	20
3	28	19	20
4	32	25	34
5	41	40	45

Tempo (em minutos) gasto por cada colaborador
para distribuir os jornais.

   Apresente **uma função recursiva** que, dados:

   - o número (positivo) J de centenas de jornais a distribuir;
   - o número (positivo) C de colaboradores; e
   - a respectiva tabela T de tempos (de dimensão J por C),

   calcula o tempo mínimo da distribuição de todos os jornais. Indique claramente o que representa cada uma das variáveis que utilizar e explicite a chamada inicial.