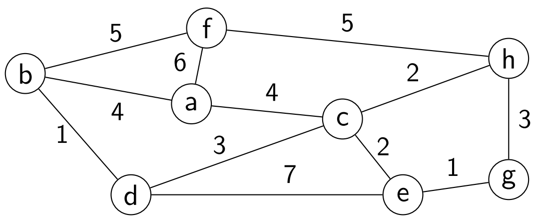
Estruturas de Dados e Algoritmos II
Exame de Recurso

Departamento de Informática
Universidade de Évora

23 de Junho de 2022

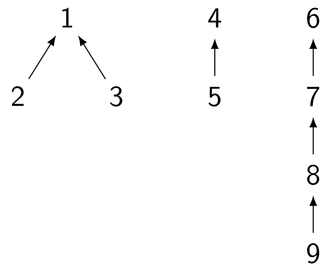
1. Considere o grafo da figura seguinte:



(a) [0,5 valores] Apresente um caminho simples de comprimento 4.

(b) [2,5 valores] Apresente uma ordem pela qual os vértices do grafo poderiam ser acrescentados à árvore de cobertura mínima, durante uma aplicação do algoritmo de Prim, a partir do vértice a.

2. Considere uma implementação do TAD Partição, com reunião por altura e compressão de caminho, e a partição do conjunto $\{1, \ldots, 9\}$ ilustrada na figura.



(a) [1 valor] Apresente o estado do vector que representa a partição da figura.

(b) [1 valor] Partindo do estado anterior, apresente o estado do vector depois da execução da operação $\mathsf{UNION}(1, 4)$.

(c) [1 valor] Partindo do estado inicial, apresente o estado do vector depois da execução da operação $\mathsf{FIND\text{-}SET}(9)$.

3. Um Percurso numa Matriz é um percurso que vai desde o canto superior esquerdo da matriz até ao canto inferior direito, andando sempre para a direita ou para baixo. O custo de um percurso é a soma dos valores em cada posição visitada durante o percurso.

Por exemplo, na matriz à direita, há seis percursos distintos desde o canto superior esquerdo, com coordenadas (1,1), até ao canto inferior direito, com coordenadas (3,3). O custo do percurso que vai para baixo, depois para a direita, depois para baixo e, finalmente, para a direita é $3+5+-5+9+-1 = 11$. O percurso que anda duas posições para a direita e, de seguida, duas para baixo é, destes percursos, o com maior custo, que é $3 + -2 + 8 + 5 + -1 = 13$.

3	–2	8
5	–5	5
–6	9	–1

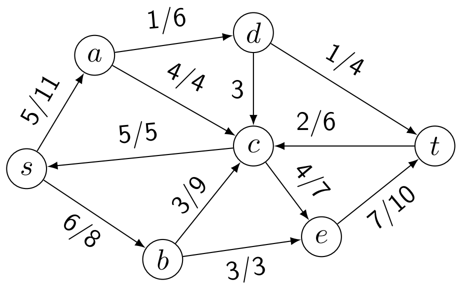
Recorrendo a grafos, como calcularia o custo máximo de um percurso numa matriz:

(a) [1,5 valores] Apresente o grafo que usaria na aplicação da sua solução à matriz acima.

(b) [0,5 valores] Como classifica o grafo que apresentou na alínea anterior?

(c) [2 valores] Descreva o algoritmo completo que utilizaria para obter o resultado pretendido, após a construção do grafo. (Pode incluir pseudo-código na sua descrição. Se o seu algoritmo recorre a algoritmos dados nas aulas, não precisa de os descrever.)

4. Considere a rede de fluxos seguinte e o fluxo f nela representado:



(a) [1,5 valores] Apresente a rede residual correspondente.

(b) [1,5 valores] Efectue uma iteração do algoritmo de Edmonds-Karp sobre a rede apresentada. Apresente o caminho considerado, a sua capacidade residual, e a representação da rede com o fluxo calculado.

5. Considere a função recursiva $d_F(i, j)$, onde:

- $F = (f_1\ f_2\ \ldots\ f_n)$, com $n > 0$, é uma sequência de inteiros positivos; e
- $1 \leq i \leq j \leq n$.

$$d_F(i,j) = \begin{cases} f_i & \text{se } i = j \\ \max\left\{\ f_i - d_F(i+1,j),\ f_j - d_F(i,j-1)\ \right\} & \text{se } i < j \end{cases}$$

(a) [3 valores] Apresente o pseudo-código de uma função iterativa que, dada uma sequência não vazia de inteiros positivos $F = (f_1\ f_2\ \ldots\ f_n)$, calcula e devolve o valor de $d_F(1, n)$.

(b) [1 valor] Estude (justificando) as complexidades temporal e espacial da sua função.

6. [3 valores] Seja T uma tabela expansível, dotada de uma operação $\mathsf{ADD}(T, x)$, com o seguinte funcionamento: se há espaço livre na tabela, o elemento x é simplesmente lá inserido; se a tabela está cheia, ela é substituída por uma nova tabela, cuja capacidade é 1,5 vezes a da tabela antiga, e todos os elementos já presentes na tabela, assim como x, são inseridos na nova tabela.

Seja T_i o estado da tabela após a i-ésima operação sobre ela, e sejam $size_i$ e num_i, respectivamente, a capacidade da tabela e o número de elementos que ela contém, no estado T_i.

Recorrendo ao método do potencial, calcule o custo amortizado da operação ADD. Use como função potencial a função:

$$\Phi(T_i) = 3\,num_i - 2\,size_i,$$

Considere que o custo real de uma operação é número de inserções de elementos efectuadas durante a sua realização, e que, no estado inicial T_0, a capacidade da tabela é 0.

Na sua resposta, assuma que, quando não há espaço disponível, a capacidade da tabela é sempre aumentada em 50%. Ignore a possibilidade de este valor ser inferior a 1 ou de a capacidade da tabela ser um número ímpar.