· U  C ·

UNIVERSIDADE DE COIMBRA
Faculdade de Ciências e Tecnologia
Departamento de Engenharia Informática

Laboratório de Programação Avançada

Exame Especial – 28 de julho de 2017

Nome: _____ Nº de estudante: _____

8 pontos no total, 3 horas, sem consulta.

1. Apresente a complexidade temporal do seguinte algoritmo recursivo que retorna o índice que um número x ocupa numa lista S ou retorna -1 se não encontrar esse número na lista. Justique a sua resposta recorrendo ao Teorema Mestre. Assuma que $S = (S[1], \ldots, S[n])$ é uma lista de n números ordenados por ordem não-decrescente, que a primeira chamada deste algoritmo é $Tsearch(S, 1, n, x)$. e que as operações aritméticas demoram tempo constante. (1 ponto)

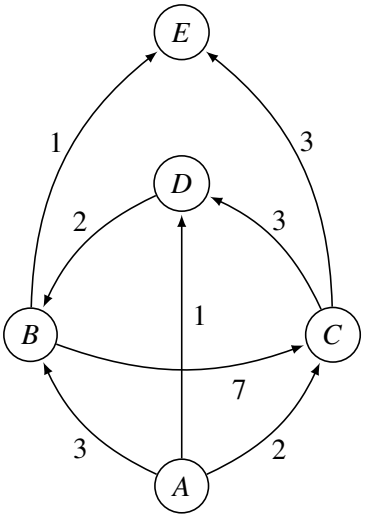
Function $Tsearch(S, \ell, r, x)$
  if $r \geq 1$ **then**
    $i = \ell + (r - \ell)/3$
    $j = r - (r - \ell)/3$
    if $S[i] = x$ **then**
      return i
    if $S[j] = x$ **then**
      return j
    if $S[i] > x$ **then**
      return $Tsearch(S, \ell, i - 1, x)$
    else if $S[j] < x$ **then**
      return $Tsearch(S, j + 1, r, x)$
    else
      return $Tsearch(S, i + 1, j - 1, x)$
  return -1

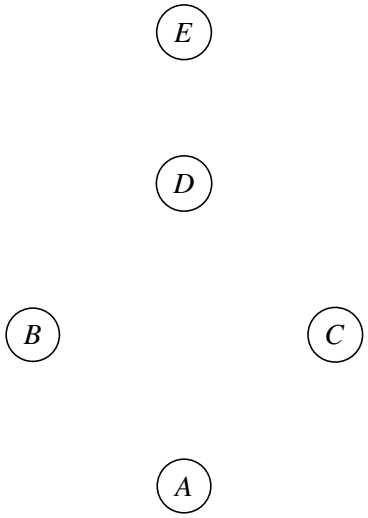
Teorema Mestre (versão geral):
Seja $a \geq 1$, $b > 1$, $d \geq 0$.

$$T(n) = \begin{cases} aT(n/b) + n^c & \text{if } n > 1 \\ d & \text{if } n = 1 \end{cases} \Rightarrow$$

$$T(n) = \begin{cases} \Theta(n^c) & \text{if } \log_b a < c \\ \Theta(n^c \log n) & \text{if } \log_b a = c \\ \Theta(n^{\log_b a}) & \text{if } \log_b a > c \end{cases}$$

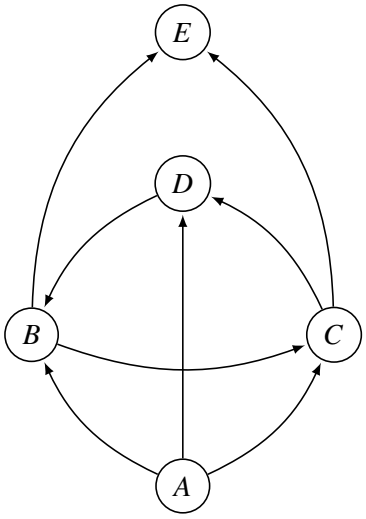
2. No seguinte grafo à esquerda existem dois caminhos mais curtos entre o vértice A e o vértice E com a mesma distância. Com base no algoritmo de Dijkstra, indique, no grafo à direita, os arcos que pertencem a esses dois caminhos e os valores da distância mais curta até cada vértice. Indique também a sequência dos vértices visitados por esse algoritmo (na caixa). (1.5 pontos)





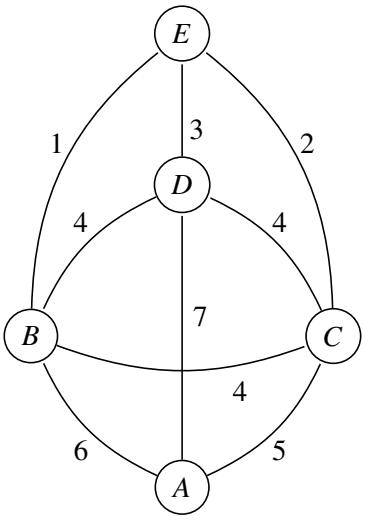
Sequência dos vértices visitados:

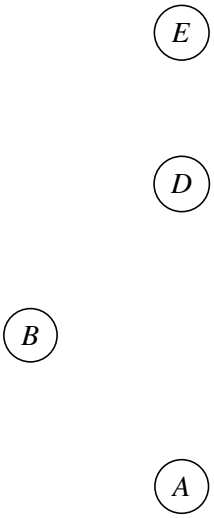
3. Encontre as componentes fortemente conexas do grafo seguinte recorrendo ao algoritmo de Tarjan. Reporte, à direita, a árvore de DFS a partir do vértice A e escolha os vértices para a travessia de acordo com a ordem alfabética das etiquetas nos vértices. Indique as componentes fortemente conexas que encontrou (na caixa). (1.5 pontos)

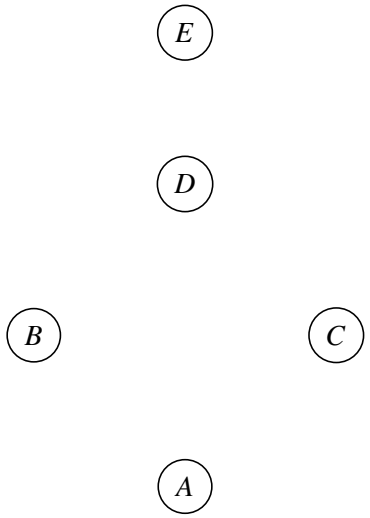


Componentes fortemente conexas:

4. Dado o grafo à direita, desenhe a árvore geradora mínima (em baixo, à esquerda) e o grafo da estrutura de dados *union-find*, sem compressão de caminho (em baixo, à direita), recorrendo ao algoritmo de Kruskal. Quando necessário, ligue a raiz da árvore com menor altura à raiz da árvore com maior altura e, em caso de empate, escolha, como raiz, o vértice que apresentar a menor etiqueta (alfabeticamente). (1.5 pontos)







5. O grafo no enunciado do exercício 4 não contém pontos de articulação. Indique, em baixo, as arestas que necessitam de ser removidas desse grafo de modo a obter pelo menos um ponto de articulação, e o(s) ponto(s) de articulação. Sobre esse grafo que considerou, reporte a árvore de DFS do algoritmo para encontrar pontos de articulação (à direita), com início no vértice *A* e escolhendo os vértices para a travessia de acordo com a ordem alfabética das etiquetas nos vértices. A classificação nesta questão é inversamente proporcional ao número de arestas removidas. (1.5 pontos)

Arestas a remover:

Ponto(s) de articulação:

6. Dado n dados, tendo cada dado m faces, e um valor inteiro S, escreva o pseudo-código de um algoritmo de programação dinâmica que retorne o número total de vezes que pode sair o valor total S quando os n dados são jogados em conjunto. Explique o raciocínio que utilizou para chegar ao algoritmo. (1 ponto)