· U · C ·

UNIVERSIDADE DE COIMBRA
Faculty of Science and Technology
Department of Informatics Engineering

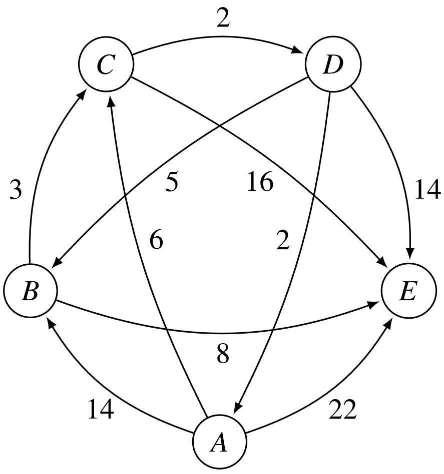
Laboratório de Programação Avançada

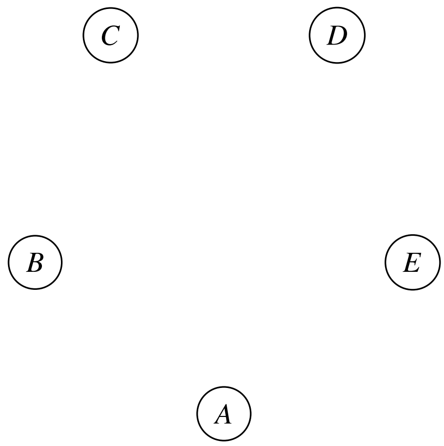
Retake Exam – July 5 2018

Name: ______________________________ Student ID: ____________________

8 grade points in total, 2h 30m, closed books.

1. Find a shortest path in the left-hand graph between vertex A and vertex E using Dijkstra algorithm. Draw the arcs that belong to the path in the right-hand graph. Fill in the table with the visited vertices, ordered according to the visiting order of Djikstra algorithm, and with the shortest distance from vertex A to every vertex. (1.5 g.p.)

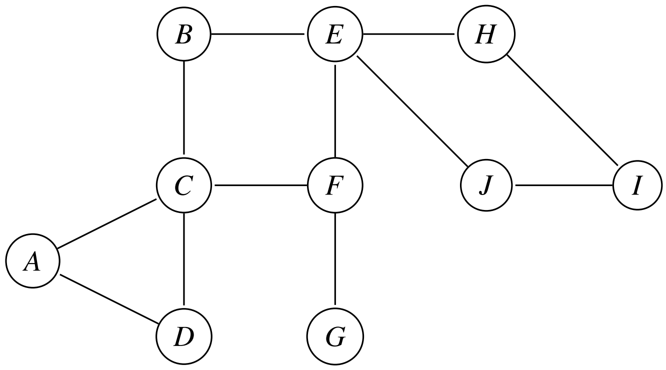




Vertices:					
Distance:					

2. Consider that the arc between vertex D and vertex A in the graph above has weight -2. Assuming that a path can contain repeated vertices, is the shortest path found in the previous exercise still valid after this transformation? Justify your answer. (1 g.p.)

3. Find the articulation points in the following graph. Justify your answer by reporting the DFS tree starting from node E, choosing the vertices for traversal in alphabetic order of the labels, and by explicitly writing the final values for `dfs` and `low` at each vertex. In addition, report the articulation points in the box below, ordered by the time they are found in the DFS tree traversal, as well as the criterion used to identify each point. (1.5 g.p.)



Articulation points:

4. Consider the following recurrence relation. Let $S_1, \ldots, S_n$ and $T_1, \ldots, T_m$ be two sequences of n and m characters, respectively. We define $A(i, j)$, $0 \leq i \leq n$, $0 \leq j \leq m$, as follows

$$A(i,j) = \begin{cases} j & \text{if } i = 0 \\ i & \text{if } j = 0 \\ \max\{A(i-1,j)+1,\ A(i,j-1)+1,\ A(i-1,j-1)+d\} & \text{if } i > 0 \text{ and } j > 0 \end{cases}$$

where $d = 1$ if $S_i = T_j$ and $d = 0$ otherwise. Give the pseudo-code of a bottom-up dynamic programming algorithm that explores the recurrence above to find the value for $A(n,m)$ and discuss its time complexity. (1.5 g.p.)

5. Write a tail-recursive function `Find(A,n)` in pseudo-code that computes the minimum element of an array A that contains n positive integers. Assume that you cannot create neither global variables nor local variables in your program. Consider that the first element in A has index 0. (1 g.p.)

6. Consider the following problem: Given a grid of size $n \times m$, count the number of paths from the top left corner to the bottom right corner of the grid, assuming that only right or down directions are allowed in the grid.

(a) Let $M(i, j)$, $1 \leq i \leq n$, $1 \leq j \leq m$ be the total number of paths that reach position (i, j) in the grid. The value of $M(i, j)$ can be found by the following recurrence relation:

$$M(i,j) = \begin{cases} 1 & \text{if } i = 1 \text{ or } j = 1 \\ M(i-1,j) + M(i,j-1) & \text{otherwise} \end{cases}$$

Show that the calculation of $M(i, j)$ for any i and j is correct. (1 g.p.)

(b) The following pseudo-code describes a bottom-up dynamic programming algorithm that solves the problem above. Consider now that some positions cannot be visited, that is, they are *forbidden*. Let function `forbidden(i,j)` return `true` if position (i, j) is forbidden and `false` otherwise. Explain how would you modify the pseudo-code below to count the total number of paths that do not go through the forbidden positions. (0.5 g.p.)

Function *Count*(n,m)
  for $i = 1$ to n **do**
    $M[i, 1] = 1$
  for $j = 1$ to m **do**
    $M[1, j] = 1$
  for $j = 2$ to m **do**
    for $i = 2$ to n **do**
      $M[i, j] = M[i-1, j] + M[i, j-1]$
  return $M[n,m]$

UNIVERSIDADE DE COIMBRA

Faculdade de Ciências e Tecnologia
Departamento de Engenharia Informática

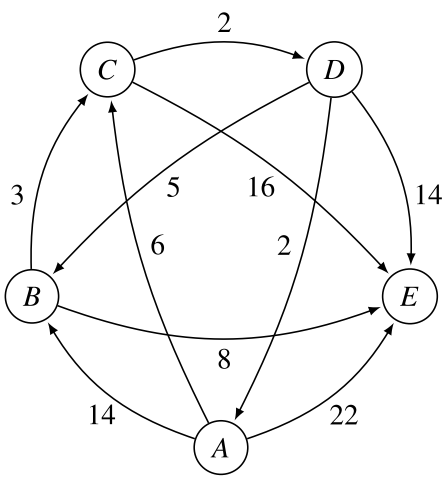
Laboratório de Programação Avançada

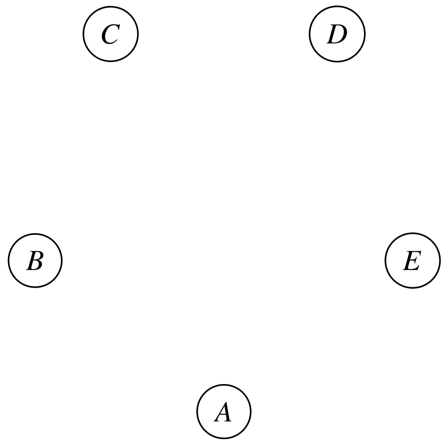
Exame de recurso – 5 de julho de 2018

Nome: ______________________________ Nº de estudante: ______________

8 pontos no total, 2 horas e 30 minutos, sem consulta.

1. Encontre um caminho mais curto no grafo à sua esquerda entre o vértice A e o vértice E com o algoritmo de Dijkstra. Desenhe os arcos que pertencem a esse caminho no grafo à sua direita. Preencha a tabela com os vértices visitados, ordenados pela ordem de visita de acordo com o algoritmo de Dijkstra, e com a distância mais curta do vértice A a cada vértice. (1.5 pontos)

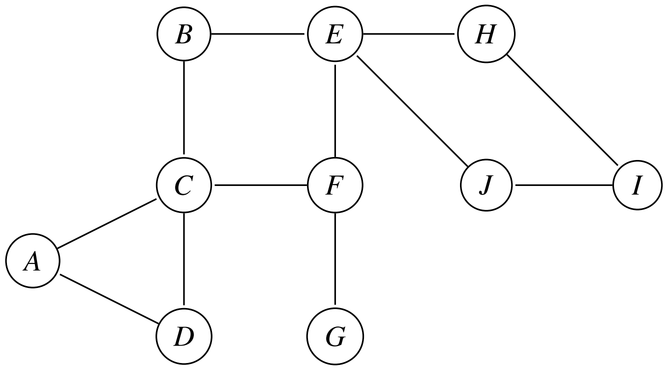




Vértices:					
Distância:					

2. Considere que o arco entre o vértice D e o vértice A no grafo acima tem um peso de -2. Assumindo que um caminho pode conter vértices repetidos, acha que o caminho mais curto encontrado do exercício anterior continua válido depois desta transformação? Justique a sua resposta. (1 ponto).

3. Encontre os pontos de articulação no grafo seguinte. Para justificação da sua resposta, reporte a árvore de procura em profundidade a partir do vértice E, escolhendo os vértices para a travessia de acordo com a ordem alfabética das etiquetas, e indique explicitamente os valores finais de `dfs` and `low` em cada vértice. Reporte igualmente os pontos de articulação na segunda caixa, ordenados pelo tempo em que foram encontrados durante a travessia em profundidade, e o critério utilizado para identificação de cada ponto. (1.5 pontos)



Pontos de articulação:

4. Considere a seguinte recorrência. Sejam $S_1, \ldots, S_n$ e $T_1, \ldots, T_m$ duas sequências com n e m caracteres, respetivamente. Define-se $A(i, j)$, $0 \leq i \leq n$, $0 \leq j \leq m$, da seguinte forma

$$A(i,j) = \begin{cases} j & \text{se } i = 0 \\ i & \text{se } j = 0 \\ \max\{A(i-1,j)+1,\ A(i,j-1)+1,\ A(i-1,j-1)+d\} & \text{se } i > 0 \text{ e } j > 0 \end{cases}$$

em que $d = 1$ se $S_i = T_j$ e $d = 0$ caso contrário. Escreva um pseudo-código de uma abordagem de programação dinâmica ascendente que explore a recorrência acima para encontrar o valor de $A(n,m)$ e discuta a sua complexidade temporal. (1.5 pontos)

5. Escreva o pseudo-código de uma função recursiva de cauda, denominada `Find(A,n)`, que calcula o menor elemento de um vetor A que contém n inteiros positivos. Assuma que não pode criar variáveis locais nem globais no seu programa. Considere que o primeiro elemento de A tem o índice 0. (1 ponto)

6. Considere o seguinte problema: Dada uma grelha de tamanho $n \times m$, conte o número total de caminhos que começam no canto superior esquerdo da grelha e que terminam no canto inferior direito. Assuma que as direções permitidas são apenas para baixo ou para a direita.

   (a) Seja $M(i,j)$, $1 \leq i \leq n$, $1 \leq j \leq m$, o número total de caminhos até à posição (i,j) na grelha. O valor de $M(i,j)$ pode ser calculado pela seguinte recorrência:

   $$M(i,j) = \begin{cases} 1 & \text{se } i = 1 \text{ ou } j = 1 \\ M(i-1,j) + M(i,j-1) & \text{caso contrário} \end{cases}$$

   Demonstre que o cálculo de $M(i,j)$ está correto para qualquer valor de i e j. (1 ponto)

   (b) O seguinte pseudo-código descreve uma abordagem de programação dinâmica ascendente que resolve o problema acima. Considere agora que algumas posições na grelha não podem ser visitadas, isto é, são *proibidas*. Assuma que existe uma função `forbidden(i,j)` que retorna `verdade` se a posição (i,j) é proibida e `falso` caso contrário. Explique como modificaria o pseudo-código para contar o número total de caminhos que não passam pelas posições proibidas. (0.5 pontos)

Function *Count*(n,m)
  for $i = 1$ to n **do**
    $M[i,1] = 1$
  for $j = 1$ to m **do**
    $M[1,j] = 1$
  for $j = 2$ to m **do**
    for $i = 2$ to n **do**
      $M[i,j] = M[i-1,j] + M[i,j-1]$
  return $M[n,m]$

6