· U C ·

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

Estratégias Algorítmicas

Exame Recurso – 28 de junho de 2024

Nome: Tiago Jorge Coimbra da Silva   Nº de estudante: 2022216213

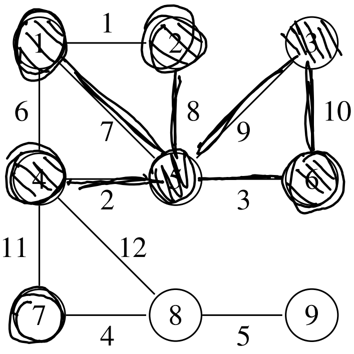
13 pontos no total, 2 horas, sem consulta.

1. Escreva o pseudo-código de um algoritmo recursivo que permita imprimir o Triângulo de Pascal (alinhado à esquerda) até um determinado valor de n, explorando a seguinte recorrência para $n > 0$ e $0 < k \leq n$: $P(n,k) = P(n-1,k-1) + P(n-1,k)$. Nota: Tem de considerar explicitamente os casos-base no seu algoritmo (2 pontos).
   Exemplo para $n = 5$:

```
1   1
2   1 1
3   1 2 1
4   1 3 3 1
5   1 4 6 4 1
```

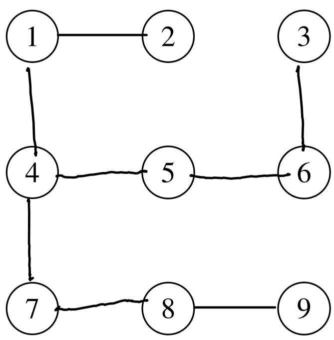
Function main() for i=1...n for k=1...i print(P(n,k)) print("\n") Return 0	Function P(n,k) If n=1 or k=1 or k=n return 1 Else Return P(n-1,k-1)+P(n-1,k)

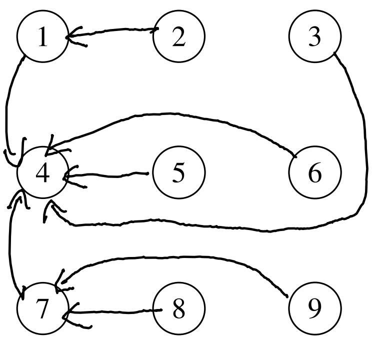
2. Considere o seguinte grafo.



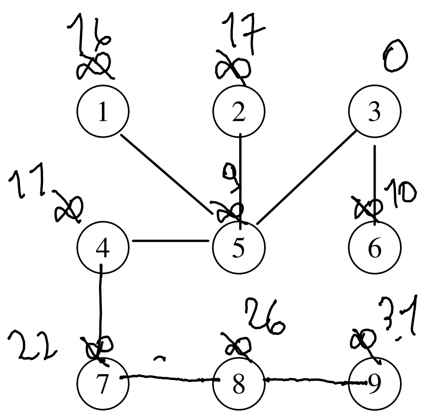
{1,2} ✓
{4,5} ✓
{5,6} ✓
{7,8} ✓
{8,9} ✓
{1,4} ✓
{1,5} ✗
{2,5} ✗
{5,3} ✗
{3,6} ✓
{4,7} ✓
{4,8} ✗

a) Desenhe a árvore geradora mínima (à esqueda) e o grafo da estrutura de dados *union-find*, sem o passo de compressão de caminho (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 etiqueta com o menor valor. (2 pontos)





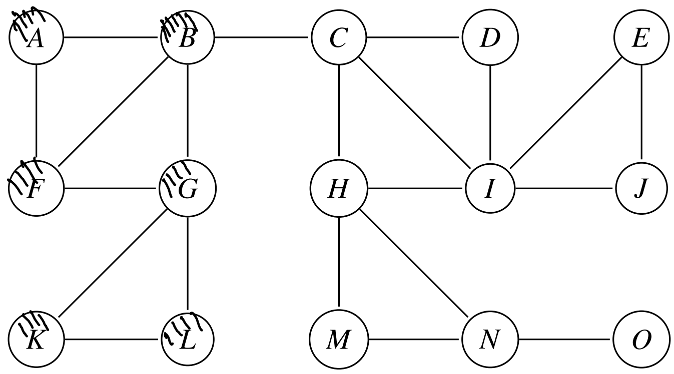
b) Desenhe o caminho mais curto do nó 3 a cada nó do grafo usando o algoritmo de Dijkstra, indicando a distância mais curta em cada nó. Assuma que pode ir em ambos os sentidos em cada aresta. No fim da página, indique os nós visitados pela ordem que o algoritmo os seleciona (2 pontos).

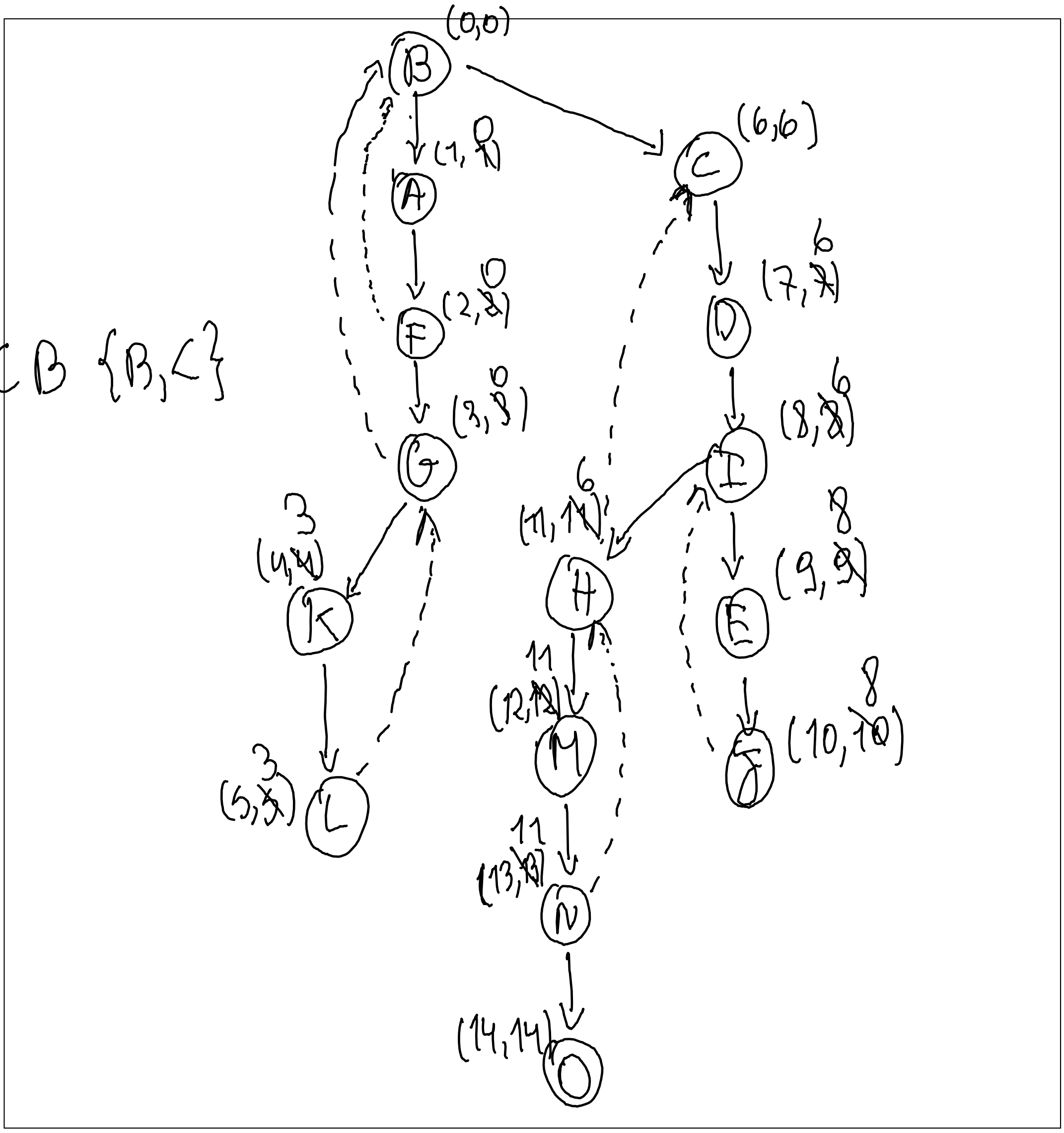


1 → ~~∞~~ 16(5)
2 → ~~∞~~ 17(5)
3 → 0
4 → ~~∞~~ 11(5)
5 → ~~∞~~ 9 (3)
6 → ~~∞~~ 10 (3)
7 → ~~∞~~ 22(4)
8 → ~~∞~~ 26 (7)
9 → ~~∞~~ 31 (8)

Nós visitados: 3, 5, 6, 4, 1, 2, 7, 8, 9

3. Encontre os pontos de articulação e as pontes no grafo seguinte. Para justificação da sua resposta, reporte a árvore de procura em profundidade a partir do vértice B, 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 e as pontes encontradas, ordenados pelo tempo em que foram encontrados durante a travessia em profundidade. (4 pontos)





GHICB {B,C}

4. Considere a seguinte recorrência para $i = 0, \ldots, n$, $j = 0, \ldots, n$ e $d_i \geq 0$:

$$D(i,j) = \begin{cases} \infty & \text{se } i < 1 \text{ or } j < 1 \\ d_i + \min\{D(i-1,j), D(i,j-1), D(i-1,j-1)\} & \text{caso contrário} \end{cases}$$

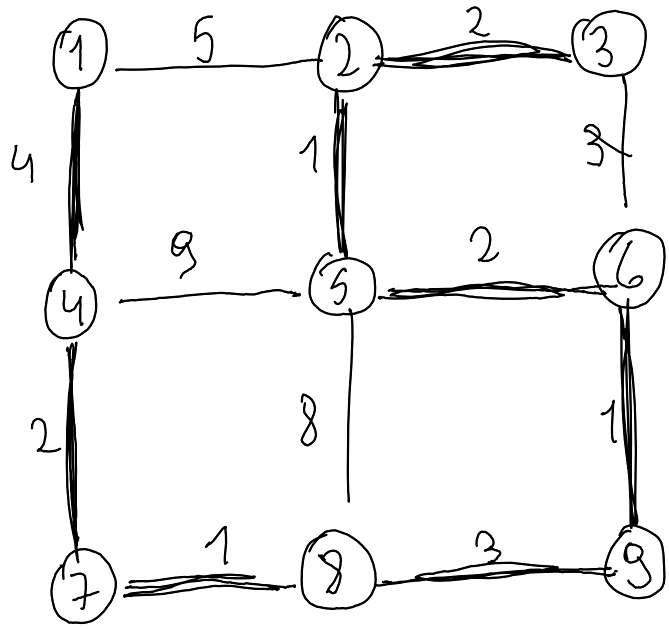
Tendo em conta que o valor que pretende obter é retornado por $D(n,n)$, apresente o pseudo-código de um algoritmo de programação dinâmica ascendente (*bottom-up*) que explore a recorrência acima para obter esse valor e discuta a sua complexidade computacional. (3 pontos)

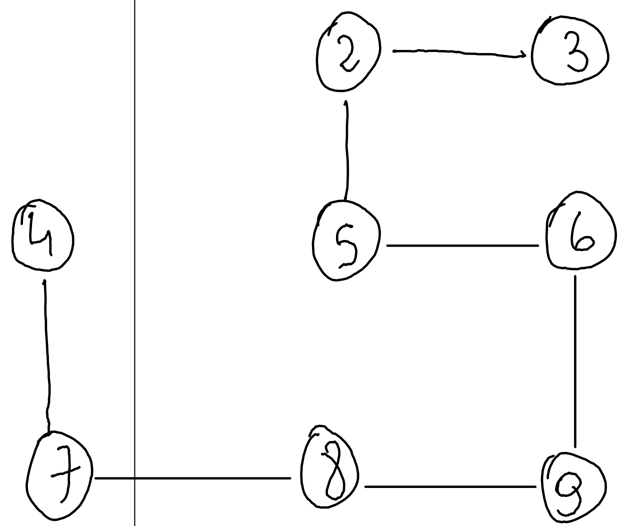
```
Function D(n)
  for i = 0 ... n
      DP[i][0] ← ∞
      DP[0][i] ← ∞

  for i = 0 ... n
      for j = 0 ... n
          DP[i][j] = 0
          m1, m2, m3 = ∞
          If i-1 > 0
              m1 = DP[i-1][0]
          If j-1 > 0
              m2 = DP[i][j-1]
          If both
              m3 = DP[i-1][j-1]
      DP[i][j] = d[i] + min{m1, m2, m3}

  return DP[n][n]
```

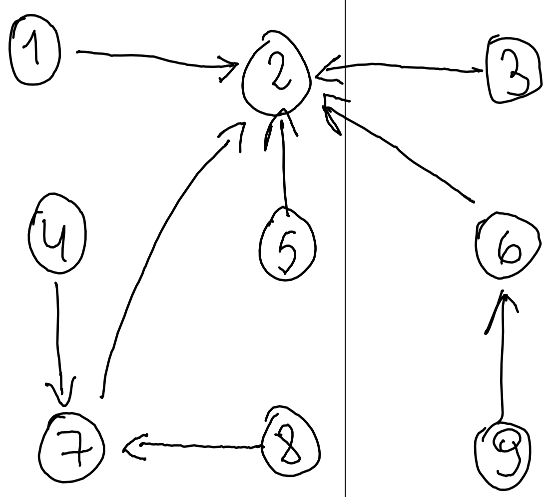
Temporal: $O(n^2)$

Espacial: $O(n^2)$

Coisa Random







6