GABARito ↓

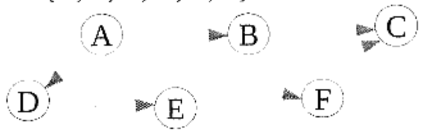
Universidade Federal Rural de Pernambuco
Departamento de Estatística e Informática
Fundamentos de Problemas Computacionais II
Prova: 2ª V.A.
Data: 23/03/2023
Prof. Tiago A. E. Ferreira

Obs.: A prova é individual e todas as questões deverão conter seus respectivos desenvolvimentos para serem consideradas!

1) **(2.0 Pontos)** Dado que seja conhecida a matriz de pesos **W** de um grafo G=(V,E) dirigido que não contém ciclos com peso resultante negativo. É possível determinar os pesos dos caminhos mais curtos com fontes múltiplas com as informações apresentadas? Se possível, como? Justifique sua resposta.

2) **(3.0 Pontos)** Como estudado, o algoritmo de Floyd-Warshall utiliza o processo de programação dinâmica para calcular os percursos mais curtos de fontes múltiplas, sendo a ideia básica para o processo a decomposição de um percurso mais curto entre os vértices $i \rightsquigarrow j$ em dois percursos mais curtos $i \rightsquigarrow k$ e $k \rightsquigarrow j$, sendo k um vértice intermediário do percurso $i \rightsquigarrow j$. Desta forma, pergunta-se: é possível de adaptar o algoritmo de Floyd-Warshall para a determinação dos percursos mais longos com fontes múltiplas para um grafo dirigido e ponderado, com a restrição de todos os pesos serem positivos? Justifique a resposta.

3) **(2.0 Pontos)** Dado grafo G=(V,E) mostrado abaixo. Observe que não são apresentados pesos nas arestas de G. Desta forma, é possível considerar que todos os pesos são iguais.
   a. **(1.0 Ponto)** É desejado de calcular os caminhos mais curtos com fontes múltiplas para o grafo abaixo. Dentre os algoritmos estudados em sala, qual o mais adequado para tal fim? Justifique a resposta.
   b. **(1.0 Ponto)** Aplique o algoritmo apresentado no item (a) e mostre uma matriz **M** com todos os pesos dos caminhos mais curtos com fonte múltiplas, de tal forma que $m_{ij} = \delta(i,j)\ \forall\ i,j \in \{A, B, C, D, E, F\}$.



4) **(3.0 Pontos)** Seja o quebra-cabeça dos 8 números. Este jogo consiste de um tabuleiro onde há 8 peças que podem correr no plano (para cima, para baixo e para os lados) dependendo da posição do "espaço vazio", mas nunca podem sair do tabuleiro. Seja o estado inicial e o estado final (solução buscada) representados na figura abaixo. Pede-se:
   a. **(1.0 Ponto)** Esquematize diagramaticamente como um grafo poderia representar o espaço de estados deste problema. Descreva o modelagem do problema, ou seja, o que representa vértices e as arestas para o dado problema.
   b. **(1.0 Ponto)** Defina um função heurística para a resolução deste problema. Verifique se a função proposta é admissível.
   c. **(1.0 Ponto)** É desejado a aplicação do algoritmo A* para a solução deste problema. Dado que $f(n) = g(n) + h(n)$, onde $h(n)$ é a função heurística e $g(n)$ é a função de custo real executado, apresente uma possível expressão para $f(n)$? Explique a expressão apresentada.

Estado inicial

7	2	4
5		6
8	3	1

Estado Final (Solução)

	1	2
3	4	5
6	7	8

Gabarito da 2ª V.A. - Fundamentos de Problemas Computacionais II

Prova realizada em 23/03/2023

1) É conhecido W (matriz de Pesos) de um grafo $G=(V,E)$ dirigido e sem ciclos com pesos resultantes negativos!

Resposta: Sim, é possível!

Como? Seja $l_{ij}^{(m)}$ o peso mínimo do percurso de i para j que contenha no máximo m arestas.

* Se $m=0$

$$l_{ij}^{(0)} = \begin{cases} 0, & \text{se } i=j \\ \infty, & \text{se } i\neq j \end{cases}$$

* Se $m \geq 1$

$$l_{ij}^{(m)} = \min\left(l_{ij}^{(m-1)}, \min_{1\leq k\leq n}\left\{l_{ik}^{(m-1)} + w_{kj}\right\}\right)$$

onde $W = w_{ij}$

Assim, é possível se computar as matrizes $L^{(1)}, L^{(2)}, \dots, L^{(n-1)}$, onde n é a quantidade de vértices do grafo G $\Rightarrow |V|=n$

Esta computação se dá pela multiplicação da matriz L pela matriz W. Sendo $L^{(1)}=W$ e $L^{(m)}=L^{(n-1)}$ $\forall m \geq n-1$

Então: Calcula-se

$$L^{(1)} = W$$
$$L^{(2)} = W^2 = L^{(1)}\cdot L^{(1)}$$
$$L^{(4)} = W^4 = L^{(2)}\cdot L^{(2)}$$
$$L^{(8)} = W^8 = L^{(4)}\cdot L^{(4)}$$
$$\vdots$$
$$L^{(2\lceil \lg(n-1)\rceil)} = W^{2\lceil \lg(n-1)\rceil}$$

2) O Alg. de Floyd-Warshall se baseia na idéia que um percurso desejado é a composição de subpercursos desejados. Ou seja,

dado um percurso mais curto entre i e j

$$i \xrightarrow{\text{Percurso mínimo}} j$$

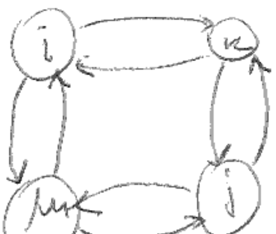
este pode ser quebrado em dois percursos, que também serão mais curtos,

$$i \xrightarrow{\text{Percurso mínimo}} k \xrightarrow{\text{Percurso mínimo}} j$$

ou seja: $i \rightsquigarrow j \equiv i \rightsquigarrow k + k \rightsquigarrow j$

Observe que se for ~~...~~ desejado o percurso mais longo tal estrutura recursiva de solução ~~...~~ não pode ser mais aplicada. Observe o contra-exemplo:

Seja G



- Sem perda de generalidade, todas as arestas têm o mesmo peso!
- O maior caminho entre i e j?

$$i \rightarrow k \rightarrow j$$

Quebrando em subpercursos!

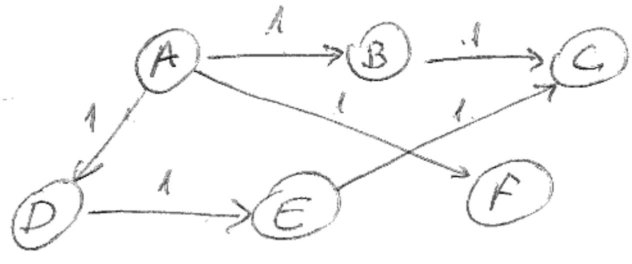
$$i \rightarrow k + k \rightarrow j$$

$i \rightarrow k$ não é mais o maior percurso entre i e k!

$k \rightarrow j$ não é mais o maior percurso entre k e j!

Assim, em geral, não é mais possível afirmar que o maior percurso entre dois vértices é composto pelos maiores subpercursos, dado um vértice intermediário!

3) Dado o grafo



Ⓐ O grafo é esparso!
Observe que $|E| = 6$ e
$|V| = 6$
logo $|E| << |V|^2$
grafo esparso

Assim, o Algoritmo indicado é

$O(V^2 \lg V + VE)$ $\longrightarrow$ o <u>Algoritmo de Johnson</u>

Ⓑ Algoritmo de Johnson (G, w)

1. compute G', onde $G'.V = G.V \cup \{s\}$, $G'.E = G.E \cup \{(s,v) : v \in G.V\}$
   e $w(s,v) = 0 \ \forall \ v \in G.V$
2. if Bellman-Ford (G', w, s) == false
   print "Erro - ciclo negativo"
   else for v in $G'.V$
   $\quad \hookrightarrow h(v) = \delta(s,v)$ // computado por Bellman-Ford

   for (u,v) in $G'.E$
   $\quad \hookrightarrow \hat{w}(u,v) = w(u,v) + h(u) - h(v)$

   Let D uma nova matriz $n \times n$

   for u in $G.V$
   $\quad \hookrightarrow$ Run Dijkstra $(G, \hat{w}, u)$ e compute $\hat{\delta}(u,v)$ $\forall v \in G.V$

   $\quad$ for v in $G.V$
   $\quad \quad \hookrightarrow d_{uv} = \hat{\delta}(u,v) + h(v) - h(u)$

   return D

Para o grafo dado $\longrightarrow$

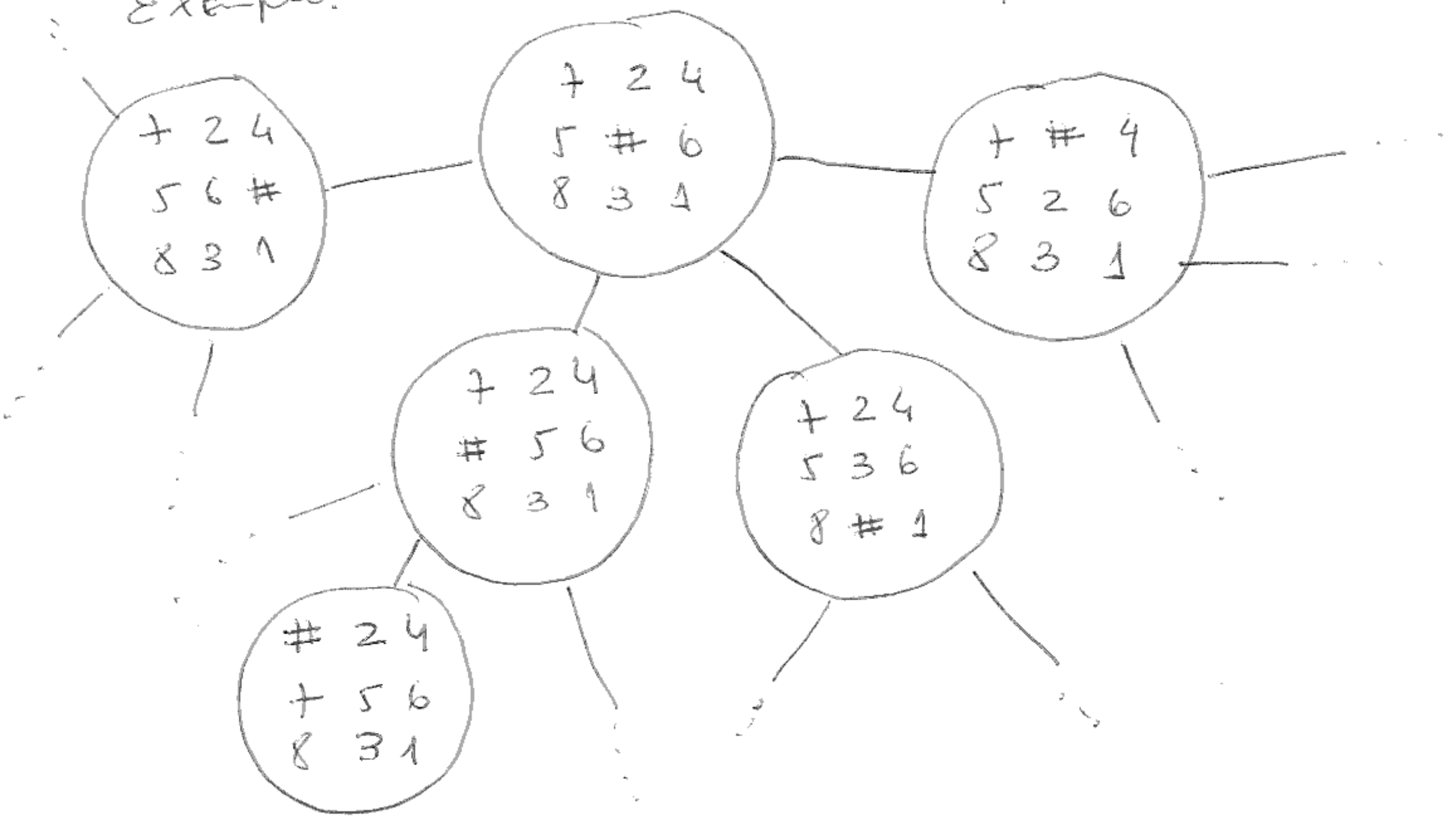
$$D = \begin{array}{c} \\ A \\ B \\ C \\ D \\ E \\ F \end{array} \begin{array}{c} \begin{array}{cccccc} A & B & C & D & E & F \end{array} \\ \begin{pmatrix} 0 & 1 & 2 & 1 & 2 & 1 \\ \infty & 0 & 1 & \infty & \infty & \infty \\ \infty & \infty & 0 & \infty & \infty & \infty \\ \infty & \infty & 2 & 0 & 1 & \infty \\ \infty & \infty & 1 & \infty & 0 & \infty \\ \infty & \infty & \infty & \infty & \infty & 0 \end{pmatrix} \end{array}$$

4) a) Vertice → Representa uma possível configuração do arranjo das peças no tabuleiro

Arestas → Transições possíveis de uma certa configuração (ou arranjo) das peça para outra configuração.

\# → Espaço vazio

Exemplo:



b) Uma possível função heurística seria a distância em número de "casas" da posição atual do número no nodo n do grafo para a posição do referido número no estado solução.

Ex:

(Estado n)	(solução)
7 2 4	# 1 2
5 # 6	3 4 5
8 3 1	6 7 8

distância p/ nº 7 é "3"

1, 2, 3 — posição esperada do nº 7

④ o mesmo p/ todos!

P/o exemplo $f(n) = 3+1+2+2+2+6+3+2$

$\boxed{f(n) = 21}$

Observe que $f(n)$ é Admissível, visto que no mínimo temos que mover a qtd de vezes.

e) Com o mesmo raciocínio do item (b), $g(n)$ é o somatório do número de movimentos de cada peça, desde o estado inicial até o estado atual n.

Representando o tabuleiro por uma matriz $i \times j$, i linhas e j colunas. Seja a posição da peça $p = \{1, 2, 3, 4, 5, 6, 7, 8\}$ no estado n dada por $(i_p(n), j_p(n))$.

Seja a posição da peça p na solução dada por (i_p^*, j_p^*)

Então

$$h(n) = \sum_p \sqrt{(i_p(n) - i_p^*)^2} + \sqrt{(j_p(n) - j_p^*)^2}$$

$$g(n) = \sum_{k=1}^{n} \sqrt{(i_p(k) - i_p(k-1))^2} + \sqrt{(j_p(k) - j_p(k-1))^2}$$

onde a posição da peça p no estado inicial é $(i_p(0), j_p(0))$ e $i_p(-1) = i_p(0)$, $j_p(-1) = j_p(0)$.

Assim,

$$f(n) = g(n) + h(n)$$

$$\boxed{f(n) = \sum_{k=1}^{n} \left\{ \sqrt{(i_p(k) - i_p(k-1))^2} + \sqrt{(j_p(k) - j_p(k-1))^2} \right\} + \sum_p \left\{ \sqrt{(i_p(n) - i_p^*)^2} + \sqrt{(j_p(n) - j_p^*)^2} \right\}}$$