Matemática Discreta II — Trabalho de Programação

Período Acadêmico Emergencial (PAE) 2020-1

Resumo

Descreve-se o projeto computacional previsto na programação da disciplina, e as regras que o regem.

1 Formação dos grupos

O presente trabalho deverá ser realizado em grupos de dois estudantes. É importante enfatizar que os dois membros do grupo deverão participar ativamente da elaboração do trabalho. A efetiva participação dos estudantes poderá ser verificada através de perguntas sobre o projeto.

2 Etapas do trabalho

O projeto é constituído por duas etapas principais:

1. Construção de um código-fonte em Scilab implementando os algoritmos satisfazendo as condições apresentadas na seção 3, intitulada[1] 'Descrição do Trabalho';

2. Redação de um relatório que disserte sobre os pontos principais dos algoritmos criados, dificuldades encontradas durante o processo de implementação do código e os resultados obtidos com a aplicação da interface gráfica em alguns exemplos. O *template* com as principais seções que o relatório deve conter está disponível no Moodle.

3 Descrição do Trabalho

3.1 Algoritmos principais em grafos

Os grafos $G = (\mathcal{N}, \mathcal{A}, \partial, \omega)$ considerados no trabalho são simples, conexos, e ponderados, onde $\omega : \mathcal{A} \mapsto \mathbb{R}$, com $\omega(a) > 0$, para todo $a \in \mathcal{A}$, é a função peso dos arcos em $\mathcal{A}$.

O trabalho computacional consiste na implementação de alguns algoritmos em grafos:

1. implementação do algoritmo de Dijkstra para obtenção do caminho mínimo entre dois nós;

2. implementação do algoritmo de busca em profundidade a partir de um nó;

3. implementação do algoritmo de busca em nível a partir de um nó.

Dos algoritmos acima, você deve programar o de Dijkstra e escolher um dos outros.

Em todos os casos o algoritmo necessita que se dê de entrada uma matriz de adjacência, A, do grafo.

Uma vez dada a matriz de adjacência, em cada vez que o algoritmo rodar a escolha dos dois nós iniciais (no algoritmo de Dijkstra), ou do nó inicial (no caso dos outros dois algoritmos), deve ser feita alestoriamente com igual probabilidade de escolha entre qualquer dos nós.

[1] A demanda apresentada na subseção 3.3.4 não é obrigatória, mas pode render dois pontos adicionais na nota do trabalho, a qual pode ultrapassar a nota máxima de dez pontos. Como anunciado, a nota do trabalho entra no cálculo da média na disciplina.

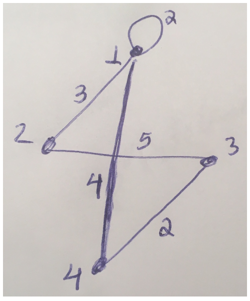
3.2 Algoritmo de representação do grafo

Há que prover um algoritmo para transformar a representação de A numa representação matricial, B, com três colunas, como apresentada no exemplo 21 da página 496 do livro-texto.

Vejamos um exemplo com a matriz de adjacência

$$A = \begin{pmatrix} 2 & 3 & 0 & 4 \\ 3 & 0 & 5 & 0 \\ 0 & 5 & 0 & 2 \\ 4 & 0 & 2 & 0 \end{pmatrix}$$

correspondente ao grafo



É interessante obter a matriz de adjacência $\bar{A}$ que informa apenas se há ou não há arco ligando um par de nós,

$$\bar{A} = \begin{pmatrix} 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix}$$

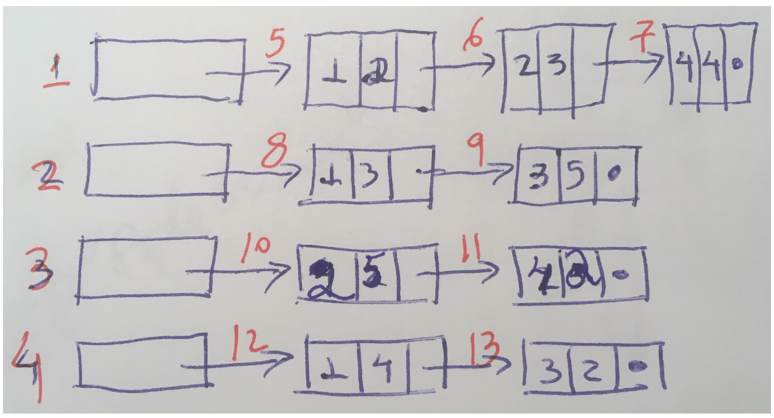
Essa matriz é a matriz de adjacência se os pesos das arestas fossem todos iguais a 1. Procure ver como calcular $\bar{A}$.

Com essa matriz podemos obter o vetor $\mathbf{a}$, 4×1, que diz quantos arcos há ligados a um nó,

$$\mathbf{a} = (3\ 2\ 2\ 2)^t$$

Não é o vetor de graus dos nós. Faça isso.

Esse grafo tem quatro nós. A matriz B, que é relacionada à representação



é, neste caso, 13×3,

$$B = \begin{array}{c} \# \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \\ 11 \\ 12 \\ 13 \end{array} \begin{pmatrix} \text{nó} & \text{peso} & \text{aponta} \\ * & ** & 5 \\ * & ** & 8 \\ * & ** & 10 \\ * & ** & 12 \\ 1 & 2 & 6 \\ 2 & 3 & 7 \\ 4 & 4 & 0 \\ 1 & 3 & 9 \\ 3 & 5 & 0 \\ 2 & 5 & 11 \\ 4 & 2 & 0 \\ 1 & 4 & 13 \\ 3 & 2 & 0 \end{pmatrix}$$

Note que as linhas de B estão relacionadas aos objetos igualmente numerados na figura. As primeiras quatro linhas de B correspondem aos nós, e as seguintes correspondem sucessivamente às entradas não-nulas de cada linha da matriz de adjacência A. Assim, a forma de construir B a partir de A tem que ser dividida em dois 'pedaços'.

Como há 3 entradas não-nulas na primeira linha de A, as linhas de 5, 6 e 7 de B dependem delas, e depois as linhas 8 e 9 dependem das duas entradas não-nulas da segunda linha de A, etc Como A tem 9 entradas não-nulas, então o número de linhas de B é 13=4+9. Note que o número de entradas não-nulas de A, 9, não é o dobro das arestas de G, uma vez que nessa contagem as arestas que vão de um nó a ele mesmo só entram uma vez. Já as arestas que ligam nós distintos contam duas vezes.

Nas linhas que correspondem aos nós, pode-se escolher -1 para substituir o * e 0 para substituir os **.

Se denotarmos por $B(i,j)$ a entrada ij da matriz B, então

$$B(1,3) = 4(\#\text{ de nós}) + 1 = 5$$
$$B(2,3) = B(1,3) + 3(\#\text{ de arcos ligados ao nó } 1) = 8$$

Note que $\#$ de arcos ligados ao nó 1 não se iguala ao grau do nó 1. Analogamente,

$$B(3,3) = B(2,3) + 2(\#\text{ de arcos ligados ao nó } 2) = 8 + 2 = 10B(4,3) = B(3,3) + 2 = 12$$

É claro que este pedaço pode ser feito em geral usando o vetor **a**.

A partir da linha 5 em diante, há que usar a informação do vetor ∂ e da matriz A.

É interessante então construir vetores que indiquem, para cada nó i, quais são as linhas que são utilizadas para representar as informações do nó na matriz B. Sabemos que isso ocorrerá na linha i, e depois numa sequência de linhas que começam em, digamos $\text{in}(i)$, e que terminam em $\text{fim}(i)$, funções de i que devem ser calculadas.

Assim, quando um elemento de A, digamos $A(i,l) \neq 0$, saberemos que a aresta (i,l), pensada ligada a i, deve ser informada em alguma linha entre $\text{in}(i)$ e $\text{fim}(i)$, digamos ind. Neste caso, se não for a última vez que $A(i,l)$ for não-nula na linha i de A, devemos ter:

$$\begin{aligned} B(\text{ind},1) &= l \\ B(\text{ind},2) &= A(i,l) \\ B(\text{ind},3) &= \text{ind}+1 \end{aligned}$$

mas se for a última vez que aparece um elemento não-nulo na linha, devemos colocar, $B(\text{ind},3) = 0$.

Use essas ideias para fazer um algoritmo que calcule a função $A \mapsto B$, onde A é uma matriz de adjacência de um grafo com pesos, simples e conexo.

E depois o algoritmo inverso, $B \mapsto A$. Note que o tamanho de A depende da linha onde ocorre o 'último' -1.

3.3 Dados, sua entrada e saída, e sua visualização

Com respeito à entrada e saída de dados tem-se as seguintes demandas, sendo que as apresentadas nas três primeiras subseções são obrigatórias, e como já dito, a da última subseção não é obrigatória, mas pode render até dois pontos extras na nota do trabalho.

3.3.1 Matriz de adjacência do Brasil

Fornecer a matriz de adjacência de um mapa do Brasil, bem como vetores **x** e **y** contendo posições de pontos representativos dos estados. Nao envolve programação. (Mas se você fizer 3.3.4, então pode utilizar para fazer esta).

3.3.2 Imagem do grafo

Dada uma matriz de adjacência, A, e vetores posição de nós, representar na tela do computador o grafo resultante através de uma imagem.

3.3.3 Menu

Dar escolha de mapas (Brasil, Inglaterra, Mercosul) em um menu e criar figura dividida em 4 zonas:

- subplot(2,2,1) o mapa da região escolhida;
- subplot(2,2,2) o grafo da região com os nós numerados;
- subplot(2,2,3) o grafo rotulado com os nomes das regiões representadas pelos nós;
- subplot(2,2,4) o grafo com os nós numerados sobreposto ao mapa da região.

O programa deve ler os mapas e as correspondentes matrizes de arquivos de dados.

- A: adjacência;
- **x**: abcissas das posições dos nós;
- **y**: ordenadas das posições dos nós;
- R: nomes das regiões (rótulos).

3.3.4 Menu iterativo

Dado um mapa, criar o grafo a partir de cliques no mapa.

- criar nós
- criar arestas
- retirar nós
- retirar arestas
- informar o peso de uma aresta (pedir de todas as arestas, quando se informar que todas as arestas foram dadas)
- criar menu para essas opções
- criar menu de escolha de mapa
- salvar as variáveis em arquivo