

## Trabalho 0 – Grau de Entrada

Data de Entrega: 21/03/2014

### Introdução

Considere uma rede social como o Twitter, onde pessoas seguem e são seguidas por outras pessoas. Tal rede pode ser modelada como um grafo orientado (dígrafo), onde os vértices são os participantes da rede e as arestas são suas relações. **Observação:** Note que o modelo de rede do Facebook, por exemplo, é um pouco diferente, pois as relações de amizade são bidirecionais.

Este trabalho consiste em a) ler um grafo do tipo Twitter apresentado de forma semelhante a uma lista de adjacências, b) realizar alguns cálculos simples e c) apresentar como saída o número de seguidores de cada participante (grau de entrada dos vértices). A especificação detalhada das entradas e saídas do programa é apresentada a seguir.

### Entrada

Os dados – fornecidos na entrada-padrão – apresentam um formato semelhante ao das lista de adjacências, conforme mostra o seguinte exemplo:

```
200
1 6 2 5 9 13 21 32
2 3 13 65 139
...
```

- ❖ A primeira linha indica o número de vértices,  $N$ , do grafo, ou seja, o número de participantes da rede social.
- ❖ Cada uma das  $N$  linhas subsequentes representa as relações de cada um dos  $N$  participantes da rede.
  - O primeiro número de cada linha – 1 e 2, no exemplo acima – representa o identificador do

participante,  $i$ .

- O segundo número de cada linha – 6 e 3, no exemplo acima – indica o número de pessoas seguidas por  $i$ ,  $S(i)$ , ou seja, o grau de saída do vértice.
- Os  $S(i)$  números seguintes representam os participantes seguidos por  $i$  (vértices destino de cada aresta).
- ❖ No exemplo acima, o participante 2 segue outros 3 participantes: 13, 65 e 139.
- ❖ Os participantes serão identificados por números inteiros de 0 a  $N-1$ . Não haverá *loops* ou arestas múltiplas na entrada.

## Saída

Como saída, devem ser apresentados, para cada participante, o seu identificador,  $i$ , seguido pelo número de seguidores do mesmo,  $E(i)$ , ou seja, o grau de entrada do vértice. Para o exemplo acima:

```
1 2
2 5
...
```

❖ **Cada linha representa um participante da rede.**

- O primeiro número de cada linha é seu identificador,  $i$ .
- O segundo número de cada linha é a quantidade,  $E(i)$ , de participantes que o seguem (grau de entrada de  $i$ ).

## Exemplo

Entrada:

```
5
0 2 1 4
1 3 0 2 3
2 1 0
3 3 1 2 4
4 1 1
```

Saída:

```
0 2
1 3
2 2
3 1
4 2
```

1  
0 0

0 0

## Outras Informações Importantes

- ❖ O trabalho deve ser feito em duplas, sendo que ambos os membros da dupla devem submetê-lo através do [Sistema de Submissão de Programas](#) (SSP) na disciplina SCC0216 – Modelagem Computacional em Grafos (2014/1) Turma: C.
- ❖ Todas as submissões são checadas para evitar cópia/plágio/etc. Portanto, evite problemas e implemente o seu próprio código.
- ❖ Comente o seu código com uma explicação rápida do que cada função, método ou trecho importante de código faz (ou deveria fazer). Os comentários serão checados e valem nota.
- ❖ Mantenha a modularização entre código e dados, ou seja, divida o código explicitamente entre estruturas de dados (grafo) e algoritmos de manipulação. A modularização será checada e vale nota.
- ❖ Entradas/saídas devem ser lidas/escritas a partir dos dispositivos de entrada e saída padrões, logo são suficientes as funções **printf()** e **scanf()**. Para testar o programa fora do SSP, pode-se usar redirecionamento de arquivos. Para isso utilize os operadores **<** e **>**, como no seguinte exemplo:

```
# ./trab0 < entrada.txt > saida.txt
```