Trabalho 4 - Redes Complexas

SCC-216 Modelagem Computacional em Grafos M. Cristina/Jorge

1 Descrição do trabalho

Data de entrega: 30 de junho de 2014 no SSP

A medida de centralidade intermediação (*betweenness centrality*) é uma medida de redes complexas que quantifica a participação de um vértice u em caminhos de comprimento mínimo.

$$betweenness(u) = \sum_{\substack{i=1 \\ (i \neq u)}}^{N} \sum_{\substack{j=1 \\ j \neq i \ j \neq u}}^{N} \frac{\sigma(i,u,j)}{\sigma(i,j)} \tag{1}$$

Em que $\sigma(i,u,j)$ é o número de caminhos de comprimento mínimo entre i e j que passam por u e $\sigma(i,j)$ é o número total de caminhos mínimos entre i e j.

Implemente o algoritmo de *betweenness centrality* descrito no paper "A Faster Algorithm for Betweenness Centrality", de Ulrik Brandes para calcular a centralidade de todos os vértices de um grafo (algoritmo na próxima página).

O algoritmo recebe como entrada um grafo não direcionado e não ponderado, e tem como saída o vetor $C_B[\,]$, a medida de centralidade para cada um dos vértices.

Algorithm 1: Betweenness centrality in unweighted graphs

$C_B[v] \leftarrow 0$, $v \in V$;
for $s \in V$ **do**
  $S \leftarrow$ empty stack;
  $P[w] \leftarrow$ empty list, $w \in V$;
  $\sigma[t] \leftarrow 0$, $t \in V$; $\sigma[s] \leftarrow 1$;
  $d[t] \leftarrow -1$, $t \in V$; $d[s] \leftarrow 0$;
  $Q \leftarrow$ empty queue;
  enqueue $s \rightarrow Q$;
  while *Q not empty* **do**
    dequeue $v \leftarrow Q$;
    push $v \rightarrow S$;
    foreach *neighbor w of v* **do**
      // w found for the first time?
      if $d[w] < 0$ **then**
        enqueue $w \rightarrow Q$;
        $d[w] \leftarrow d[v] + 1$;
      end
      // shortest path to w via v?
      if $d[w] = d[v] + 1$ **then**
        $\sigma[w] \leftarrow \sigma[w] + \sigma[v]$;
        append $v \rightarrow P[w]$;
      end
    end
  end
  $\delta[v] \leftarrow 0$, $v \in V$;
  // S returns vertices in order of non-increasing distance from s
  while *S not empty* **do**
    pop $w \leftarrow S$;
    for $v \in P[w]$ **do** $\delta[v] \leftarrow \delta[v] + \frac{\sigma[v]}{\sigma[w]} \cdot (1 + \delta[w])$;
    if $w \neq s$ **then** $C_B[w] \leftarrow C_B[w] + \delta[w]$;
  end
end

2 Critério de Avaliação

Dados:

$N_{acertos}$ = Número de casos de teste corretos;

N_{casos} = Número total de casos de teste;

$T4$ = Nota do trabalho 4.

$T4 = \frac{N_{acertos} * 10}{N_{casos}}$

3 Entrada

A entrada deve ser lida do teclado (stdin).

A entrada contém apenas um caso de teste, composto por várias linhas.

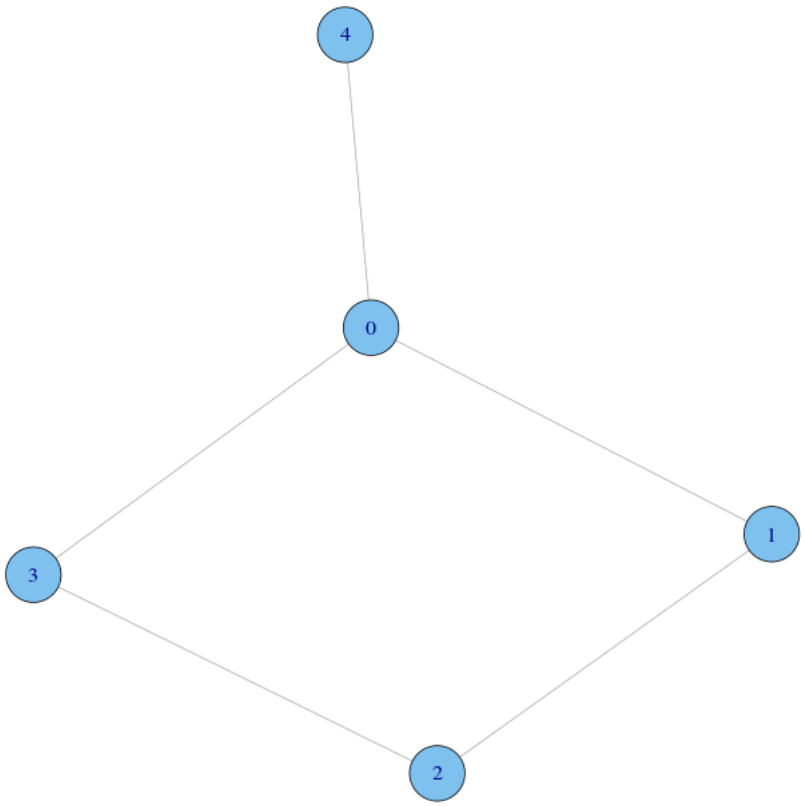
A primeira linha de entrada contém dois números inteiros, V e A, que indicam respectivamente o número de Vértices e o número de Arestas do grafo. $1 \leq V \leq 100$. $0 \leq A \leq 1000$.

As A linhas seguintes contém dois números inteiros, $V1$, $V2$, vértices que devem ser conectados no grafo (de $V1$ para $V2$). $0 \leq V1, V2 \leq 99$.

4 Saída

A saída contém a medida *betweenness centrality* para cada vértice, separados por quebra de linha, com uma casa decimal. A formatação deve ser **printf("%.1f")**.

5 Exemplo



5.1 Entrada

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

5.2 Saída

3.5
1.0
0.5
1.0
0.0

6 Plágio

ATENÇÃO: O sistema SSP utiliza a ferramenta MOSS, de Stanford, para detecção de plágio. Para mais informações, visite: <http://theory.stanford.edu/ aiken/moss/>.