PUC-Rio — Departamento de Informática
Prof. Marcus Vinicius S. Poggi de Aragão
Período: 2014.2 — 3 de novembro de 2014
Data da Entrega: 25 de novembro de 2014

ESTRUTURAS DE DADOS AVANÇADAS (INF 1010)

2º Trabalho de Implementação

Descrição

A entrega do trabalho consiste de:

- OBRIGATÓRIO: Um e-mail para poggi@inf.puc-rio.br com ASSUNTO (ou SUBJECT) EDA142T2 contendo os arquivos correspondentes ao trabalho. O NÃO ENVIO DESTE E-MAIL, COMO SOLICITADO, IMPLICA QUE O TRABALHO NÃO SERÁ CONSIDERADO.
- Um documento contendo o roteiro de desenvolvimento dos algoritmos (e dos códigos), os itens pedidos, comentários e análises sobre a implementação e os testes realizados(papel).
- A impressão dos trechos RELEVANTES dos códigos fonte (papel).
- O trabalho pode ser feito em grupo de 2 ou 3 alunos.

Este trabalho prático consiste em desenvolver códigos para representar grafos e executar algoritmos para identificar elementos e características destes grafos e, principalmente, analisar o desempenho das implementações destes algoritmos com respeito ao tempo de CPU. O desenvolvimento destes códigos e a análise devem seguir os seguintes roteiros:

- Descrever os algoritmos informalmente.
- Demonstrar o entendimento do algoritmo explicando, em detalhe, o resultado que o algoritmo deve obter e justificá-lo.
- Apresentar as tabelas contendo as características obtidas, além dos tempos de execução utilizados pelos algoritmos sobre as instâncias testadas.
- Documente o arquivo contendo o código fonte de modo que cada passo do algoritmo esteja devidamente identificado e deixe claro como este passo é executado.
- Para a medida de tempo de CPU das execuções utilize as funções disponíveis no link correspondente na página do curso, um exemplo de utilização é apresentado. Quando o tempo de CPU for inferior à 5 segundos, faça uma repetição da execução tantas vezes quantas forem necessárias para que o tempo ultrapasse 5 s (faça um while), conte quantas foram as execuções e reporte a média.

- Obrigatoriamente apresente as soluções obtidas, ou uma caracterização destas soluções, no caso de serem muito extensas.

- Obrigatoriamente apresente tabelas contendo uma coluna para cada algoritmo aplicado às instâncias, com o tempo de CPU utilizado. Cada linha da tabela é associada a uma instância e contém a identificação da mesma. Nesta tabela coloque as instâncias em ordem crescente de tamanho.

A corretude código será testada sobre um conjunto de instâncias que será distribuido. A descrição das instâncias estão em anexo aos arquivos disponibilizados.

Considere as definições a seguir:

- ***Subgrafo Induzido***: Dado um grafo $G = (V, E)$, não orientado, onde V e E são os conjuntos de vértices e de arestas de G, respectivamente, sendo $V = \{1, 2, \ldots, n\}$ e n o número de vértices do grafo. Um conjunto $S \subset V$ induz o seguinte subgrafo: $G(S) = (S, E(S))$ onde $E(S) = \{(u, v) | u \in S,\ v \in S, \text{ and } (u, v) \in E\}$.

- ***Clique***: Dado um grafo $G = (V, E)$, não orientado, o subgrafo induzido pelo conjunto $S \subseteq V$ $G(S)$ é uma clique se $G(S)$ é um grafo completo e para todo $w \in V \setminus S$, $G(S \bigcup \{w\})$ não é um subgrafo completo.

- ***Partição em Cliques***: Dado um grafo $G = (V, E)$, não orientado, uma partição em cliques é um conjunto de conjuntos S_1, S_2, ..., S_k cujos grafos induzidos $G(S_1)$, $G(S_2)$, ..., $G(S_k)$ são cliques, $S_1 \bigcup S_2 \bigcup \ldots \bigcup S_k = V$ e a interseção de quaisquer dois conjuntos $S_p \bigcap S_q$ é vazia.

- ***Cobertura Minimal por Cliques***: Dado um grafo $G = (V, E)$, não orientado, uma partição em cliques é um conjunto de conjuntos S_1, S_2, ..., S_k cujos grafos induzidos $G(S_1)$, $G(S_2)$, ..., $G(S_k)$ são cliques, $S_1 \bigcup S_2 \bigcup \ldots \bigcup S_k = V$ e a exclusão de qualquer dos conjuntos S_1, ..., S_k faz com que $S_1 \bigcup \ldots \bigcup S_k \subset V$.

- ***Coloração***: Dado um grafo $G = (V, E)$, não orientado, sendo $V = \{1, 2, \ldots, n\}$ e n o número de vértices do grafo, e um inteiro K, determinar uma atribuição $c(v) : V \longrightarrow \{1, \ldots, K\}$, tal que para toda aresta $(u, w) \in E$, $c(u) \neq c(w)$.

- ***Clique Máxima***: Dado um grafo $G = (V, E)$, determinar um conjunto $S \subset V$, que corresponda a uma clique, cuja cardinalidade seja a maior possível, i.e. máxima. Este número é representado por $\omega(G)$.

- ***Número Cromático***: Dado um grafo $G = (V, E)$, o seu número cromático $\chi(G)$ é o menor valor de K para o qual uma coloração existe.

- Explique porque $\omega(G) \leq \chi(G)$ para todo G.

Existem inúmeras partições em cliques para um grafo dado. Presentemente, não se conhece algoritmo de complexidade melhor que exponencial no tamanho do grafo para se determinar $\omega(G)$ ou $\chi(G)$ quando $\omega(G) < \chi(G)$.

O objetivo desse trabalho é obter informações sobre a estrutura dos grafos distribuidos. Para isso, serão executados algoritmos para encontrar cliques e particionar grafos em cliques, além de algoritmos para encontrar uma coloração para os grafos.

Naturalmente, encontraremos cliques de cardinalidade menores que $\omega(G)$ e colorações que utilizam mais do que $\chi(G)$ cores.

Os problemas que serão tratados são: Clique Máxima, Número Cromático e Cobertura Minimal por Cliques. Abaixo são descritos algoritmos com o objetivo de trazer informação sobre esse problemas.

Algoritmos a implementar:

1. *encontra_clique*:
   (a) Inicie com S, um conjunto vazio
   (b) Seleciona um vértice de V de grau máximo em $G = (V, E)$
   (c) Adicione a S um vértice que seja vizinho a todos os vértices já em S dentre estes, escolha o vértice de maior grau.
   (d) Se nenhum vértice puder ser adicionado a S, Pare! S é uma clique.
2. *encontra_cobertura_minimal*:
   (a) Inicie com $V' = V$. Faça $q = 1$.
   (b) Encontre uma clique em $G(V')$ correspondente ao conjunto S'_q.
   (c) Faça $V' \longleftarrow V' \setminus S'_q$.
   (d) Se $V' \neq \emptyset$. Faça $q \longleftarrow q + 1$ e volte ao passo **(b)**
   (e) Caso contrário, para p de 1 a q faça:
      i. Adicione a S'_p um vértice que seja vizinho a todos os vértices já em S'_p dentre estes, escolha o vértice de maior grau.
      ii. Se nenhum vértice puder ser adicionado a S'_p, Pare! S'_p é uma clique.
      iii. Faça $S_p = S'_p$.
   (f) O conjunto $S_1, \ldots, S_q$ é uma *cobertura minimal em cliques.*
3. *encontra_coloração*:
   (a) Selecione um vértice $v \in V$.
   (b) Execute uma *Busca em Profundidade* em $G = (V, E)$. A partir de v. Utilize a marcação dos vértices para atribuir as cores. Faça $c(v) = 1$. Marque cada vértice visitado com a cor de menor índice possível.
   (c) Ao final indique o maior índice utilizado (o K da coloração).

Pode-se utilizar variações dos algoritmos propostos. Estas teriam o objetivo de sempre buscar encontrar a maior clique possível e o menor número de cores possível em uma coloração.

Os resultados devem reportar para cada grafo do conjunto distribuido as seguintes informações:

- Tamanho da maior clique encontrada
- Número mínimo de cores encontrado para colorir o grafo
- Número de cliques de cada tamanho encontrados na partição. Se várias partições forem encontradas, reportar a média, mínimo e máximo desses valores.

Se o tempo de CPU, que também deve ser reportado, permitir, execute muitas vezes os algoritmos aleatorizando as escolhas dos vértices. Isso permitirá encontrar maiores cliques, menos cores, e indicar quão bem definidas são as possíveis partições em cliques dos grafos distribuidos.

ESTE TRABALHO TEM COMO MAIOR OBJETIVO OS EXPERIMENTOS COM OS ALGORITMOS Portanto, apresente um documento que seja O MAIS DETALHADO POSSÍVEL no relato dos experimentos e na análise dos seus resultados.