



Trabalho Prático – Minimum Edge Clique Covering Problem

Grupos: duplas, indicadas apropriadamente (verifique com sua dupla se ela fará o TP ou não);

Avaliação: Código-fonte comentado, resultados corretos para instâncias disponibilizadas, desempenho (vide tabela de estratos).

| Estrato | Desempenho | Nota |
|---------|------------|------|
| 1 | Melhor | 110% |
| 2 | 10% | 100% |
| 3 | 10%-15% | 85% |
| 4 | 15%-25% | 75% |
| 5 | 25%-35% | 70% |
| 6 | 35%-50% | 65% |
| 7 | >50% | 30% |

Pontos extras: 10% de acréscimo para o código de melhor desempenho;

Data de entrega: 19 de junho, até as 10:00 via *e-mail*. Trabalhos atrasados ou incompletos não serão aceitos.

Enunciado

Este trabalho consiste na implementação de um método para solução do problema de cobrimento mínimo de arestas por cliques (ou *minimum edge clique covering problem*). O problema consiste em cobrir todas as arestas de um grafo usando o número mínimo possível de cliques.

Um **clique** é uma estrutura em grafos definida por um *subgrafo completo*. Diz-se que um clique *cobre* as arestas pertencentes a ele. Note que uma aresta pode ser coberta mais de uma vez, ou seja, por mais de um clique, sem prejuízo à solução do problema. O número mínimo de cliques necessários para cobrir todas as arestas de um grafo G (número de cobertura de cliques) é denotado por $cc(G)$. Este problema é NP-difícil e até mesmo métodos heurísticos de última geração podem ter tempo de execução proibitivo ou obter soluções de baixa qualidade.

A Figura 1 ilustra a cobertura das arestas de um grafo por quatro cliques. Cada clique é delimitado pela área azul com bordas pontilhadas.

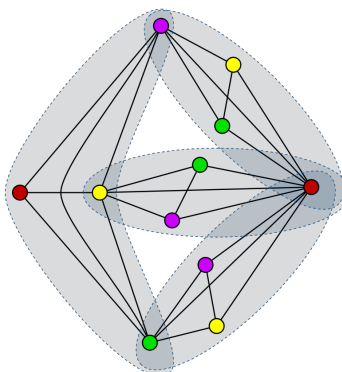


Figura 1. Grafo com arestas cobertas por quatro cliques.

A Figura 2 ilustra a cobertura das arestas de um grafo por vários cliques, porém, com sobreposição de cliques, ou seja, uma mesma aresta é coberta mais do que uma vez. Cada clique é delimitado pela área com bordas vermelhas.

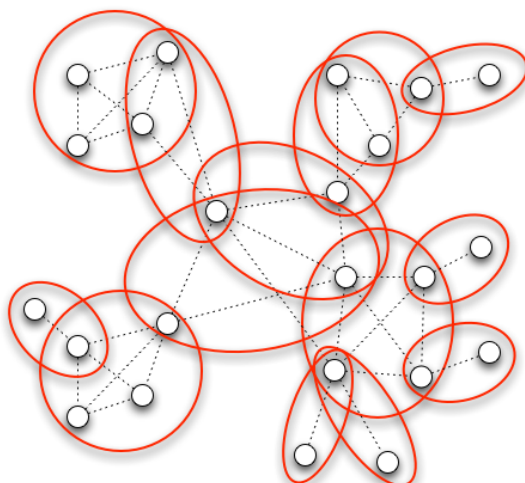


Figura 2. Grafo com arestas cobertas por vários cliques, com sobreposição.

Formato da entrada

São disponibilizadas instâncias para testes. Cada arquivo possui informações que a respeito de um único grafo e obedece ao seguinte formato:

- Número de vértices (n)
- Número de arestas (m)
- Descrição das arestas (e)

Exemplo

```
p edge 5 4
e 1 4
e 3 4
e 5 3
e 2 3
```

Formato da Saída

Os programas devem imprimir ao final, em uma linha, o número mínimo de cliques agentes necessários e o tempo de execução. Em um **arquivo**, deve ser informado para cada clique, os vértices que o compõem.

Abordagens

Exceto pelas restrições indicadas na seção abaixo, serão aceitas quaisquer abordagens, ficando a avaliação condicionada à apresentação do trabalho, em que os alunos deverão demonstrar domínio sobre a abordagem empregada.

Restrições

Não são permitidos recursos computacionais como programação paralela e distribuída, entretanto, a escolha da linguagem de programação e a utilização de outros recursos da própria linguagem são livres.

Não são permitidos recursos de programação matemática (Programação Linear, Programação Inteira, etc.).

O algoritmo deve ser determinístico, ou seja, a resposta para uma determinada instância deve ser sempre a mesma, além de o código não “chutar” a resposta. Algoritmos não determinísticos serão desconsiderados.

Também haverá restrições quanto ao tempo de execução do algoritmo.

Entregáveis

Código fonte criado (um único arquivo, sem bibliotecas ou projetos), e relatório com os resultados obtidos para as instâncias. Não é necessário incluir no relatório a descrição do método ou código fonte.

Máquinas para Testes

Serão tomadas como referência as máquinas do laboratório de ensino COM30 no sistema operacional Ubuntu. Portanto, os resultados reportados devem ser referentes àqueles computadores.

Como Medir o Tempo?

Em C++, adicione o trecho de código abaixo para medir o tempo. Recomenda-se medir o tempo depois da leitura dos dados e após o término da geração da solução.

```
#include <chrono>
using namespace std::chrono;

int main(){

    duration<double> time_span2;

    high_resolution_clock::time_point t1 = high_resolution_clock::now();
    //gerar a solução aqui
    high_resolution_clock::time_point t2 = high_resolution_clock::now();
    duration<double> time_span = duration_cast<duration<double> >(t2 - t1);

    return 0;
}
```

Duplas

FABRICIO CESAR TOFANI
ANA CRISTINA LOPES DE OLIVEIRA

GABRIEL BICALHO FERREIRA
RODRIGO JULIANO DA SILVA

PAULA FRANCA TOLEDO
VALMIR RODRIGUES BUENO JUNIOR

LETICIA OLIVEIRA REZENDE
JEAN PIERRE DIAS DA SILVA

JOELINGTON DE ALMEIDA BERNARDO
SUZAN MILENA TEIXEIRA GOMES

VIVYANN FERNANDES CEDRAZ
MATEUS DOS SANTOS MOREIRA DE OLIVEIRA

JULIA EDUARDA MIRANDA DE SOUSA
JOAO VITOR GONCALVES DA SILVA

STEFANO NOGUEIRA DE AZEVEDO

SAMMUEL RAMOS DA SILVA

NICOLAS VASCA GALINDO
JOAO PEDRO SIQUEIRA MENDES

GUILHERME FERREIRA ROCHA
GUSTAVO LUCAS MOREIRA

GUSTAVO PRESOTI SALES BRITO
DANIEL PINTO MOREIRA

NICOLLAS CESAR ALCANTARA
RICARDO GABRIEL MOREIRA

IGOR GONCALVES GABARRA LOPES
ALEXSANDRO NUNES DE OLIVEIRA

CAMILLO RINALDI BARBOSA
FREDERICO LUIZ MARTINS DE SOUSA

ALIFER DUARTE SILVA
LEONARDO ISAAC SILVA FLORES

PEDRO ALVES EUZEBIO
LUCAS DE SOUZA SILVA

ANA LETICIA CHAVES NEIVA
MICHAEL LINCONL FUJIHARA MOURA