



Trabalho Prático II – O Problema de *Clique Máximo*

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%

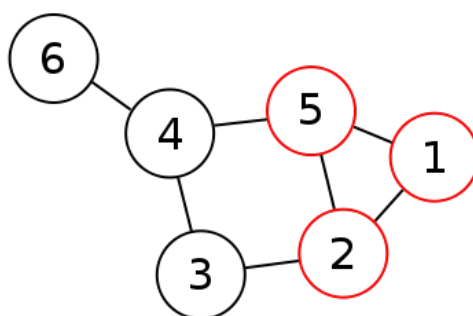
Pontos extras: 10% de acréscimo para o código de melhor desempenho;

Data de entrega: 09 de agosto, 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 Clique Máximo. O problema é definido como: dado um grafo não direcionado $G=(V, A)$, determine um subgrafo completo tal que sua cardinalidade (em número de vértices) seja a máxima.

A Figura abaixo ilustra um grafo em que a solução gerada para o problema de *Clique Máximo* é ressaltada por vértices vermelhos.



Formato da entrada

São disponibilizadas 8 instâncias para testes. Cada arquivo possui informações que obedecem ao seguinte formato:

- Dados do grafo no formato “p edge n m”, em que n representa o número de vértices e m representa o número de arestas;
- Lista de m arestas no formato “e $V_i V_j$ ”, em que V_i e V_j são os índices de dois vértices, indicando a existência de uma aresta que liga o vértice V_i ao vértice V_j ;

- EOF

Exemplo

```
p edge 4 6
e 1 2
e 1 3
e 1 4
e 2 3
e 2 4
e 3 4
```

EOF

Formato da Saída

Os programas devem imprimir ao final, em uma linha, a cardinalidade da clique máxima obtida e os vértices que a 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.).

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

Joaquim e Leonardo Luiz
Augusto e Marina
Maycon e Melina
Ana Luiza e Gabriel
Matheus e Thiago
João e Manoel
Bernardo e Igor
Anderson e Leonardo Isaac
Everton e Diego
Daniela e Ricardo
Luiz e Vinícius
Wander e Danielle