



Universidade Federal de Ouro Preto – UFOP
Instituto de Ciências Exatas e Biológicas – ICEB
Departamento de Computação – DECOM
Disciplina: Otimização em Redes
Professor: Marco Antonio M. Carvalho



Trabalho Prático II – O Problema de Caminho Mais Curto Com Restrições de Recursos

Grupos: duplas (graduação) e individual (pós);

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;

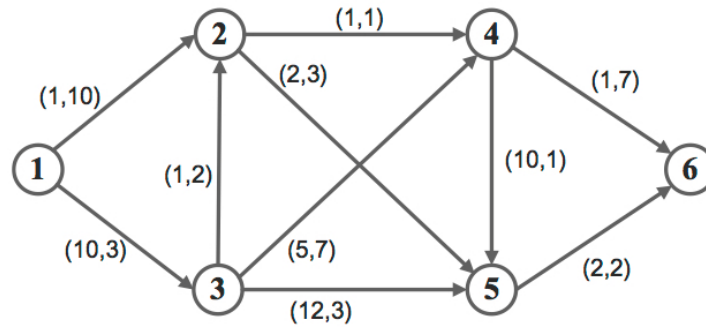
Data de entrega: 25 de junho até as 08:10, via *e-mail*. Trabalhos atrasados não serão aceitos.

Enunciado

Este trabalho consiste na implementação de um método para solução do problema de **Caminho Mais Curto Com Restrições de Recursos** (ou *Resource Constrained Shortest Path Problem*, RCSP). Informalmente, este é o problema de um viajante com um orçamento de vários recursos que tem que alcançar um determinado destino o mais rápido possível dentro das restrições de recursos impostas pelo orçamento.

Formalmente, no existe um grafo direcionado com um vértice de *origem* e um vértice de *destino*, e cada arco possui um custo e um vetor de pesos especificando seus consumos a partir de um conjunto de recursos com capacidades finitas. É procurado um caminho direcionado da origem-destino de custo mínimo, de modo que o consumo total dos arcos de cada tipo de recurso não exceda a os limites do recurso.

Considere o exemplo abaixo em que há um grafo com 6 vértices. O vértice 1 é a origem e o vértice 6 é o destino. Nos arcos, são apresentados o custo e o consumo de cada recurso diferente, cujo limite inferior é 0 e o limite superior é 10. O caminho 1-2-4-6 possui peso 3 e consome 18 unidades do recurso, sendo inviável. O caminho 1-3-2-5-6 possui peso 15 e consome exatamente 10 unidades do recurso, sendo este o caminho mais curto.



Formato da entrada

São disponibilizadas 24 instâncias para testes preliminares. Cada arquivo possui o seguinte formato:

- Número de vértices (n)
- Número de arcos (m)
- Número de recursos (K)
- Para cada um dos recursos, o seu limite inferior
- Para cada um dos recursos, o seu limite superior
- Para cada vértice, o consumo de cada recurso ao incluir o vértice no caminho
- Para cada arco, o vértice de origem, o vértice de destino, o custo do arco e a quantidade de recursos consumidos pelo arco.

Um conjunto de testes surpresa será utilizado na hora da apresentação do trabalho.

Exemplo

```
6 10 1
0
10
0
0
0
0
0
0
0
1 2 1 10
1 3 10 3
2 4 1 1
2 5 2 3
3 2 1 2
3 4 5 7
3 5 12 3
4 5 10 1
4 6 1 7
5 6 2 2
```

Formato da Saída

Os programas devem imprimir ao final, o custo total do caminho mais curto e os vértices selecionados, em uma mesma linha.

Abordagens

Serão aceitas quaisquer abordagens, ficando a avaliação condicionada à apresentação do trabalho, em que os alunos deverão demonstrar domínio sobre a abordagem empregada e desde que não sejam utilizados pacotes ou bibliotecas com implementações prontas.

Relatório

O relatório deve descrever sucinta e textualmente o método desenvolvido, sem códigos fonte ou tópicos de implementação. Adicionalmente, deve haver uma tabela com o resultado obtido para cada instância, contendo somente o custo de cada casamento obtido.

Restrições:

Haverá restrições quanto ao tempo de execução do algoritmo.

Entregáveis

Código fonte criado e relatório.

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

Bernardo e David
Túlio e Guilherme
Douglas e Melina
João e Lincoln
Renan e Joaquim
Augusto e Pedro
Victor
Robson
Ronaldo