

MC558: Projeto e Análise de Algoritmos II - 2s2022

Tarefa de Programação

Prof^o Flávio K. Miyazawa
PED: Elisa Dell'Arriva

Instituto de Computação - UNICAMP

Haroldo agora quer ser gamer!

1 Descrição do Problema

Haroldo está vidrado em um jogo eletrônico e quer sua ajuda para elaborar estratégias vencedoras. O jogo consiste em uma caça a tesouros. O ambiente é um labirinto no qual estão espalhados tesouros que devem ser coletados por um grupo de ambiciosos caçadores. Nesse labirinto, existem certos pontos de parada que estão conectados por passagens unidirecionais. Cada ponto de parada tem no máximo um tesouro e os caçadores devem explorar o ambiente de ponto em ponto. Infelizmente, andar pelo labirinto é custoso: cada passagem tem um custo que deve ser pago. Além disso, os caçadores não podem andar por uma mesma passagem, ou seja, uma vez que uma passagem foi percorrida por um caçador, nenhum outro pode passar por lá. Felizmente, essa restrição não se aplica aos pontos de parada, ou seja, um caçador pode passar por um ponto que outro caçador já visitou. Durante seu percurso, cada caçador vai acumulando uma riqueza: quando ele coleta um prêmio, é recompensado com o valor do mesmo e, cada vez que usa uma passagem, deve pagar uma taxa de uso.

Haroldo recebe um grupo de caçadores e sua função é encontrar posições iniciais para cada um deles de forma que todos cheguem ao ponto de saída do labirinto sem se encontrarem, coletando pelo menos uma certa quantidade de tesouros e acumulando o máximo valor de riqueza possível. Atenção: cada caçador deve começar em um ponto de partida diferente. Sua missão é ajudá-lo a determinar tais posições e responder o valor do total de riqueza acumulada. Para isso, você recebe as seguintes informações: o número n de pontos de paradas, o número m de passagens e, para cada uma delas, uma tripla (u, v, w) dizendo a origem, o destino e o custo da passagem, respectivamente. Para saber quais pontos de parada possuem prêmios, você recebe uma lista (p_1, \dots, p_n) tal que, para todo ponto $i = 1, \dots, n$, se o ponto i possui tesouro, p_i é seu valor e, caso contrário, $p_i = 0$. Finalmente, você recebe um inteiro k indicando o número de caçadores, um inteiro t indicando o ponto de saída do labirinto e, um inteiro P indicando a quantidade mínima de tesouros que precisam ser coletados. Essas informações são dadas no formato apresentado a seguir.

```
n m
u1 v1 w1
⋮
um vm wm
p1 ... pn
P k t
```

Todos os valores fornecidos são inteiros e tem-se que $2 \leq n \leq 1000$, $1 \leq m \leq 500000$, $1 \leq k \leq 1000$. A seguir, apresentamos um exemplo.

Entrada:

```
4 3
1 4 50
2 4 10
3 4 50
100 50 150 0
2 2 4
```

Saída esperada:

```
150
```

Observações técnicas

- O código deve ser feito em C++.
- A correção dos testes não será feita pelo SuSy, mas sim pela posterior execução local de seu código. Assim, é de extrema importância que seu código compile de acordo com o Makefile fornecido. Para correção, seu código será executado em ambiente Linux 64bits. Dado que todos os alunos podem ter acesso a máquinas com essa configuração nos laboratórios do IC, é razoável recomendarmos que você verifique se seu programa funciona nesse ambiente antes de entregar. Se estiver inseguro, guarde uma captura de tela da execução bem sucedida do programa em sua máquina. As data e hora da execução devem estar explicitadas na captura de tela, caso contrário, será difícil argumentar a seu favor se a compilação do seu programa falhar no momento da correção.
- É fornecido um arquivo template, no qual a leitura da entrada bem como a impressão da saída já estão implementadas. Além disso, o template contém uma função chamada `prizedKpaths`, a qual recebe os dados de entrada como parâmetros e deve retornar o valor encontrado pelo seu algoritmo.

2 Instruções para o Relatório

Você obrigatoriamente entregar um relatório sobre o desenvolvimento da tarefa. Para fins de clareza, o relatório deve ser estruturado em três seções:

- 1 Formulação do Programa Linear Inteiro
- 2 Descrição da Implementação
- 3 Análise de Desempenho

Na primeira seção, descreva a sua formulação do PLI, justificando as variáveis de decisão e explicando o papel de cada restrição. Dê especial atenção a essa parte, pois a qualidade da sua formulação terá significativo impacto na nota do relatório.

Na segunda seção, apresente de maneira sucinta e objetiva como sua implementação está organizada. A ideia é facilitar a leitura do seu código; exemplos de informações relevantes são: tipo e nome de variáveis principais (não faz sentido descrever variáveis auxiliares e contadores, por exemplo); quais, se alguma, funções auxiliares criou, seus nomes e propósitos. Seja objetivo. Não copie e cole seu código no relatório!

Na terceira seção, apresente uma breve análise do desempenho do seu algoritmo. O Gurobi imprime várias informações interessantes que podem vir a ser convenientes. Baseando-se no tempo de execução, indique a instância para qual seu algoritmo obteve melhor desempenho e a instância para qual ele obteve o pior desempenho.

3 Avaliação

A nota da tarefa será composta pela combinação da nota de sucesso nos testes, NS , com a nota do relatório, NR . No total, serão 20 casos de teste, dos quais apenas 10 serão disponibilizados para livre acesso antes da entrega. A nota final da tarefa, NT , é calculada da seguinte forma:

$$NT = 0,6NS + 0,4NR.$$

Note que você não precisa se limitar aos testes fornecidos para verificar a corretude de seu programa. Você é livre para criar outros casos de teste que julgue convenientes, porém eles não serão considerados na avaliação.

A identificação de qualquer tentativa de plágio implicará em $NT = 0$ para todos os envolvidos.

4 Avisos importantes

- O relatório deve ser submetido no classroom e **o nome do arquivo submetido deve ser os dígitos numéricos do seu ra**.
- Em caso de múltiplas submissões do relatório no classroom, somente a última será considerada para avaliação.
- O código deve ser submetido no SuSy. A página da disciplina no SuSy é <https://susy.ic.unicamp.br:9999/mc558a>.
- Para submeter no SuSy, utilizem somente os dígitos numéricos do RA e a senha da DAC.
- Espere, no mínimo, 60 segundos entre uma submissão e outra no SuSy.
- O número máximo de submissões no SuSy é 5 e somente a última será considerada para avaliação.
- **O prazo de entrega desta tarefa é 11/12/2022 às 23h59.** Submissões com atraso não serão consideradas para avaliação.