Pontifícia Universidade Católica de Minas Gerais
ICEI - Curso de Ciência da Computação
Disciplina: Projeto e Análise de Algoritmos
Prof. Alexei Machado

Trabalho de Implementação 2

Valor:27 pontos
Data: 23:00 horas de 12/06/2023 pelo Canvas. **Atenção: Para este trabalho não se admite atraso!**

Descrição

Problemas de otimização são bastante comuns em vários setores produtivos. Um exemplo de aplicação é o planejamento do transporte de cargas entre pontos geográficos distintos. Considere uma rede de lojas que precisa distribuir produtos entre suas *n* filiais. Ela possui um caminhão que sai da matriz e deve passar uma única vez em cada loja para entregar e/ou buscar produtos, retornando vazio à matriz. Cada loja pode ser a origem de zero ou mais produtos a serem entregues, mas é o destino de no máximo um produto. O caminhão pode carregar no máximo *k* produtos de uma só vez (k será um parâmetro alterável na interface).

Neste trabalho você deverá implementar um algoritmo que solucione o problema da coleta e entrega de produtos gastando a menor quantidade possível de combustível. O gasto de combustível é proporcional à distância total percorrida pelo caminhão que tem rendimento de 10km/litro e ao número de produtos que carrega (uma unidade de produto diminui o rendimento em 0.5 km/litro). Cada loja é identificada por um número de 1 a *n*, e sua localização dada por coordenadas *x,y* no intervalo de 0 a 500 quilômetros. As informações sobre as lojas devem ser lidas de um arquivo texto *lojas.txt* contendo uma loja por linha. Cada linha contém o número da loja, coordenada x, coordenada y e lista de zero ou mais lojas de destino de entrega, separados por espaços. Todos os valores são números inteiros. Se uma loja não for origem de nenhuma entrega, não haverá lista. A primeira linha contém a posição da matriz que é a loja 0 e não será origem nem destino de qualquer produto. Exemplo:

```
0 50 0
1 50 150 2 3
2 150 150
3 200 180 4
4 220 200
5 300 150 9
6 100 250
7 180 220 6
8 150 300
9 220 220 7 8
```

1. Escreva uma solução por força bruta para gerar as soluções possíveis e escolher a que gasta menor quantidade de combustível, sem violar as restrições quanto à carga máxima do caminhão. O programa deve mostrar de forma gráfica, em animação, a sequência das viagens do caminhão e a variação da sua carga entre as filiais.

2. Escreva uma solução por branch-and-bound, a partir da solução por força-bruta, que elimine ramos da árvore de soluções que se mostrarem com custo maior que o já obtido até aquele

momento. Verifique se o resultado é o mesmo que o obtido por força-bruta e compare os tempos de processamento das duas soluções.

3. Documente as soluções e os testes, na forma de um relatório técnico em formato PDF, segundo o padrão da PUC (site biblioteca) ou da SBC, contendo as seguintes seções:

    a) Introdução: Descrever de maneira geral o objetivo do trabalho.
    b) Solução proposta: Descrever de forma resumida os algoritmos usados para a solução do problema e analisar a sua ordem de complexidade.
    c) Implementação: Descrever detalhes dos programas implementados, principalmente aqueles utilizados para melhorar a eficiência da solução e como se organiza a interface gráfica. Descrever o sistema computacional utilizado nos testes.
    d) Relatório de testes: Descrever os testes realizados e os resultados, mostrando como ficou a sequência de lojas em cada caso, gasto de combustível e o tempo de execução.
    e) Conclusão: Discutir os resultados obtidos, comparando as soluções por força-bruta e branch-and-bound, quanto à sua ordem de complexidade e tempo de execução.
    f) Bibliografia segundo o padrão ABNT.

Considerações gerais e critérios de avaliação

1. O trabalho deverá ser feito em grupos de 2 ou 3 alunos, sem qualquer participação de outros grupos e/ou ajuda de terceiros. Cada aluno deve participar ativamente em todas as etapas do trabalho. Incentiva-se o uso do Github, com repositório privado.

2. A codificação do trabalho deve ser feita em linguagem Python, Java, C ou C++. Não poderão ser utilizados versões, bibliotecas gráficas ou qualquer recurso que não estejam instalados oficialmente nos laboratórios do ICEI.

3. Os trabalhos (código e relatório) devem ser postados por um único componente, na forma de um arquivo compactado com formato ZIP, com **tamanho máximo de 5MB**, e seu nome deve conter o nome completo de pelo menos um componente. **Os arquivos fontes devem ser copiados para o diretório raiz** e devem conter o nome de **todos** os componentes do grupo no início do código.

4. **Trabalhos iguais, na sua totalidade ou em partes, copiados, "encomendados" ou outras barbaridades do gênero, serão severamente penalizados. É responsabilidade do aluno manter o sigilo sobre seu trabalho, evitando que outros alunos tenham acesso a ele. No caso de cópia, ambos os trabalhos serão penalizados, independentemente de quem lesou ou foi lesado no processo.**

5. Durante a apresentação, poderão ser feitas perguntas relativas ao trabalho, as quais serão consideradas para fim de avaliação. Todos os componentes devem comparecer e serem capazes de responder a **quaisquer perguntas e/ou alterar e reescrever o código de qualquer parte do trabalho**. A avaliação será individual.

6. A avaliação será baseada nos seguintes critérios:

    - Correção, robustez e eficiência dos programas
    - Conformidade às especificações

- Clareza e estilo de codificação (comentários, endentação, escolha de nomes para identificadores, parametrização)
- Relatório de testes
- Relatório sobre a participação de cada componente, ou relatório de commits do Github
- Apresentação individual