

Problema

Considere um conjunto de regiões em uma cidade, compostas por vários pontos de interesse. A demanda de cada região deve ser atendida por um único ponto que faça parte dela, e cada ponto pertence a apenas uma região. A situação brevemente descrita nesse contexto está presente na roteirização de embarcações de transporte marítimo, planejamento de atendimento a pacientes na área da saúde, coleta de lixo urbano, e até em robótica. Sim: é verdade! Alguns robôs ou veículos autônomos possuem tarefas a serem executadas que exigem a visita a um único ponto de uma região, para que a transmissão de dados seja feita a outros pontos via o ponto visitado. Essas regiões podem ser, nesse contexto, áreas com alcance de redes de sensores sem fio.

O nosso estudo de caso consiste em uma empresa de entregas cujo objetivo é atender a demanda de cada região pertencente a um conjunto de regiões, cada uma composta por consumidores, através de exatamente um consumidor da respectiva região. Para tanto, considerando um número limitado de V veículos com capacidade C (cada um), encontre uma roteirização de cada veículo que saia e retorne do único depósito, e minimize o custo total de deslocamento entre os consumidores, considerando todos os veículos.

Tarefas

Tarefa 1: Extração dos Dados

A partir da descrição do contexto a ser resolvido, o grupo deverá extrair os dados de entrada do problema, identificar parâmetros e construir uma função de leitura dos dados. Os problemas para teste serão disponibilizados no Campus Virtual.

Tarefa 2: Implementação de um Algoritmo

Utilizando seus conhecimento em Algoritmos em Grafos, desenvolva um algoritmo que encontre uma solução viável para o problema descrito. **Algoritmos de enumeração completa ou força bruta não serão considerados como resposta.**

Relatório

Um relatório de no máximo 4 páginas (sem incluir a capa) deve ser confeccionado, com as seguintes informações:

- Introdução: descrição do problema e apresentação das ideias gerais do algoritmo desenvolvido.

- Algoritmo: apresentação do pseudocódigo implementado e justificar porque essa estratégia foi adotada.
- Testes computacionais: o grupo deverá testar o algoritmo implementado com todas as instâncias propostas para o trabalho. Para medir a qualidade das soluções, pede-se que seja calculado o desvio da solução de cada instância i de seu algoritmo ($sol[i]$) em relação à melhor solução conhecida de i ($bkv[i]$, presente no arquivo bkv), dado por:

$$\frac{sol[i] - bkv[i]}{bkv[i]} \times 100 \quad (1)$$

Os grupos também deverão apresentar os tempos computacionais gastos por seu algoritmo.

- Conclusão: indicar o que foi aprendido, as dificuldades e qual componente do algoritmo levou a um bom desempenho do método.

Premiação

Os trabalhos que estiverem com boa organização de código e obtiverem a maior quantidade de instâncias resolvidas com o desvio mais próximo do melhor conhecido, ganharão um bônus na nota final. Segue a relação da premiação extra do trabalho:

- Primeiro lugar geral: 5 pontos de bônus na média final do semestre;
- Segundo lugar geral: 3 pontos de bônus na média final do semestre;
- Terceiro lugar geral: 2 pontos de bônus na média final do semestre;

O grupo que ficar em Primeiro lugar geral ganhará um prêmio adicional, apresentado em sala no dia da explicação do trabalho prático.

Em caso de empate, o docente priorizará o grupo com: (i) a melhor média de desvio das instâncias; (ii) menor tempo computacional para resolver todos os exemplos. Casos omissos serão tratados diretamente pelo docente.

Atenção: os grupos que sofrerem alguma penalização por não seguirem alguma regra do trabalho estarão sujeitos a serem excluídos da premiação extra.

Regras

1. O grupo será formado por **no mínimo 3** e **no máximo 4** alunos.

2. Um único membro do grupo deverá enviar um e-mail do docente, com o nome de seus componentes, até o dia **20/05/2019**.
3. Todos os códigos poderão ser feitos em C++ ou Python3.
4. Os grupos deverão apresentar sua implementação em um único arquivo, cujo nome deve seguir o seguinte padrão: "nomeMembroSubmeteTrabalhoCV-MatriculaDesseAluno.py"(se for em Python). Exemplo: "mayron-123456.py". O nome do membro deve estar com todas as letras minúsculas.
5. Como foi exigido apenas um arquivo, sejam claros e objetivos no código.
6. A execução do programa seguirá esse padrão (exemplo em Python, para o aluno Mayron): "python mayron-123456.py -in problema1 -out mayron-123456-resultados". O arquivo "nomeMembroSubmeteTrabalhoCV-MatriculaDesseAluno-resultados" terá os resultados do algoritmo, com o seguinte padrão: "nomeProblema valorSolucao tempoExecAlgoritmo". Como exemplo, se no problema1 meu algoritmo obtém 150 de resposta em 10s, minha resposta será: "problema1 150 10"(sem aspas!). Em cada teste executado, o arquivo deverá agregar mais informações, e nunca apagar as informações dos testes já executados.
7. A execução do programa poderá, ainda, conter uma opção de geração de uma imagem e de rotas detalhadas para cada veículo, referentes a solução do problema. Nesse caso, para o exemplo anterior, basta que na linha de comando, o usuário indique -img nomeMembroSubmeteTrabalhoCV-MatriculaDesseAluno-img.png -sol nomeMembroSubmeteTrabalhoCV-MatriculaDesseAluno-sol". Exemplo: "python mayron-123456.py -in problema1 -out mayron-123456-resultados -img mayron-123456-img.png -sol mayron-123456-sol", gerará uma imagem com a solução do problema de nome "mayron-123456-img.png" e um arquivo de nome "mayron-123456-sol", que contenha as rotas para cada veículo da seguinte forma:

```
1 2 5 6
1 3 4
1 7 8
```

Nesse caso, temos três veículos e cada linha, tenho o percurso de cada um (para 8 clientes). Não é necessário indicar o retorno no vértice de origem (depósito, vértice 1).

8. A geração das imagens em grafos podem ser feitas utilizando os pacotes **IGraph** ou **Networkx**. Esses pacotes só podem ser utilizados para essa finalidade.
9. O uso de qualquer framework é permitido apenas para a geração gráfica da solução.
10. O docente agendará um horário para que o grupo apresente o trabalho pessoalmente. A entrevista fará parte do processo avaliativo.

11. O grupo que não seguir alguma regra será penalizado em ao menos 1 ponto por cada infração cometida.
12. **Data de entrega: 22/06/19, às 23h55.**
13. Qualquer cópia ou plágio acarretará em nota nula da atividade para todos os membros do grupo, e consequências em órgãos competentes da UFLA.