Trabalho Prático 2
O Problema do Caixeiro Viajante

1 Introdução

Você é um consultor e precisa visitar clientes em diferentes cidades, felizmente você e seus clientes são ricos e você tem um helicóptero que te permite viajar em linha reta entre qualquer par de clientes. Você é rico mas não é bobo então você deseja viajar gastando a menor quantidade de combustível (e tempo). Você precisa visitar todos os clientes e depois voltar para a sede de sua empresa de onde você vai partir.

2 Especificação

Neste trabalho você deverá criar um programa para resolver o problema do caixeiro viajante, definido a seguir.

Problema 1 *Problema do Caixeiro Viajante: Dado um grafo $G = (V, E)$, com um conjunto V de n vértices e um conjunto E de arestas, e um custo c_e para cada $e \in E$. Encontrar um circuito hamiltoniano (um circuito que passa exatamente uma vez por todos os vértices) que tenha custo mínimo.*

Nesse trabalho trabalharemos com um grafo completo, ou seja, existe uma aresta entre todos os pares de vértices, as distâncias são métricas e simétricas, ou seja, a distância de u para v é a mesma distância de v para u (note que existem na literatura versões do Caixeiro Viajante que podem ser diferentes).

Seu programa deverá receber dois parâmetros, o primeiro é o nome do arquivo de instância, e o segundo é nome do arquivo de saída, que deverá ser sempre igual ao nome da instância seguido de *.sol*. Você deverá ler a instância, e deverá imprimir no arquivo de saída apenas o valor da sua solução e uma lista de n vértices indicando a ordem em que são visitados, não precisa repetir o primeiro e o ultimo vértice (embora você precise considerar o custo do ultimo para o primeiro).

A entrada será da seguinte forma, um inteiro n dizendo o número de vértices, e depois n linhas cada uma com a coordenada (x_v, y_v) de cada vértice $v \in V$. Você deverá ler as coordenadas como double ou float, porém o custo entre os vértices u e v será um inteiro calculado da seguinte forma:

$$c_{(u,v)} = c_{(v,u)} = \left\lfloor \sqrt{(x_u - x_v)^2 + (y_u + y_v)^2} \right\rfloor$$

Formato da entrada:

---

n
$x_0\ y_0$
$x_1\ y_1$
$\dots$
$x_{n-1}\ y_{n-1}$

Formato da saída:

---

obj
$p_0\ p_1\ \dots p_{n-1}$

---

em que p_0 é o vértice de onde parte o caixeiro, p_1 é o primeiro vértice a ser visitado, p_2 o segundo, e assim por diante até p_{n-1}, o último vértice a ser visitado, e depois o caixeiro volta para p_0. Não precisa imprimir p_0 de novo, mas lembre-se de considerar o custo dessa viagem.

O que é fornecido

Você receberá um arquivo ".zip"com a seguinte estrutura:
tp01.zip:
data/ (uma pasta com as instâncias de entrada)
sols/ (uma pasta que contêm scripts auxiliares e onde você colocará suas soluções)
solve_all.py (um script para ajudar a executar sua solução)
corrigir.py (um script para ajudar a verificar sua solução, se você usar linux)
corrigir_win.py (um script para ajudar a verificar sua solução, se você usar windows)
Meu_Resolvedor_Python.py (um exemplo de um resolvedor em Python)
Meu_Resolvedor_C++.cpp (um exemplo de um resolvedor em C++)

O que entregar

Você deverá entregar um arquivo .zip contendo os arquivos que estão na pasta sols/, o seu código fonte e os arquivos de soluções (todos na raiz do zip, sem pastas). Sugiro que você coloque tudo na pasta sols/ depois abra ela, selecione todos os arquivos e gere o zip.

3 Instruções

- **GUARDE O SEU CÓDIGO.** A plataforma run.codes só salva sua ultima submissão, então caso você faça uma submissão piore a sua nota, perto da data de entrega você deverá re-submeter o envio que teve a nota melhor.
- Essa parte do trabalho é individual.
- Você pode usar a linguagem que desejar, são fornecidos exemplos triviais em python3 e c++, porém não está restrito a essas linguagens.
- Todos os Scripts python estão em Python 3.
- Discussões e consultas em alto nível são permitidas e encorajadas.
- Não é permitido usar códigos que não sejam seu, exceto os que são fornecidos junto com o trabalho. Plágios, Fraudes, tentativas de burlar o sistema receberão nota zero na disciplina.
- Recomenda-se que não compartilhe ou procure códigos prontos.
- Nesse trabalho não deve ser utilizado resolvedores próprios de TSP. Mas pode-se usar resolvedores de métodos gerais.
- Caso você note algum problema com a plataforma run.codes que você ache que não seja falha no seu arquivo, avise o professor o mais rápido possível.

4 Nota

O trabalho vale de 0 a 10, calculado pela média das instâncias, que são pontuadas de 0 a 10 da seguinte forma:

- Soluções inválidas ou inviáveis recebem nota 0;
- Soluções viáveis recebem $\frac{1}{3}$ da nota;
- Soluções boas (que ultrapassem um certo valor) recebem $\frac{2}{3}$ da nota;
- Soluções ótimas ou muito boas (que ultrapassem um certo valor) recebem 10;