Universidade Federal Rural de Pernambuco
Departamento de Estatística e Informática
Bacharelado em Sistemas de Informação

PROJETO FLYFOOD

HÉLDER ALVES DE MIRANDA

Recife
Fevereiro de 2024

1. Objetivos

O objetivo principal do projeto flyfood é implementar um algoritmo que seja capaz de obter a melhor rota passando por “n” pontos e retornando ao ponto de partida, simulando um problema do mundo real utilizando um algoritmo computacional. Na primeira fase será utilizado um algoritmo de força bruta e posteriormente será implementado o algoritmo do vizinho mais próximo.

1.2 Objetivos específicos

- Implementar o algoritmo de força bruta em Python;
- Realizar testes com diferentes tamanhos de matrizes e com quantidades distintas de pontos, para medir a eficiência do algoritmo;
- Registrar o tempo de execução para esses diferentes tamanhos e quantidades de pontos.
- Implementar o algoritmo do vizinho mais próximo e repetir os testes e registros realizados com o algoritmo de força bruta.
- Com a posse dos dados, realizar análise comparativa entre os dois algoritmos implementados.
- Elaborar conclusão com base nos resultados encontrados e teoria envolvida.

2. Procedimento

2.1. Configurações da máquina utilizada para implementação e testes.

O computador utilizado na implementação e teste de ambos algoritmos possui um processador Intel® Core™ i7-8550U @ 1.80GHz. O sistema operacional contou com 8Gb de memória ram DDR4 de 2130MHz, HDD de 915Gb e um SSD 120Gb. O sistema operacional instalado no SSD da máquina de teste é o Windows 10, e antes dos testes foram fechados todos aplicativos (exceto o leitor de código, VScode) e programas em segundo plano.

Para ambos os algoritmos foram utilizadas três matrizes de exemplo, cada uma executada 3 vezes e calculada a média de tempo de execução após essas 3 execuções. A Matriz pequena[4x5] com 5 pontos, matriz média[5x6] com 7 pontos e a matriz grande[6x7] com 9 pontos.

2.2 Força Bruta

O código lê um arquivo .txt e armazena suas informações em uma lista. Em seguida, cria um dicionário para mapear as coordenadas dos pontos. Identifica pontos de interesse e os armazena. Com o auxílio da biblioteca itertools, gera todas as permutações desses pontos. Em um loop, adiciona origem e destino a cada permutação, calcula a distância total e armazena o menor percurso encontrado. O menor percurso e sua distância é mostrado na saída do programa. Obtendo assim o melhor caminho possível.

Após as 3 execuções para cada matriz, foram obtidas as seguintes medias de tempo para o algoritmo de força bruta:

Matriz pequena[4x5] com 5 pontos: 0,00099730 segundos - distância: 14 dronômetros

Matriz media[5x6] com 9 pontos: 0,07156467 segundos - distância: 20 dronômetros

Matriz grande[6x7] com 11 pontos: 11,646458 segundos - distância: 28 dronômetros

2.3 Vizinho mais próximo

Esse código utiliza boa parte da estrutura do algoritmo de força bruta, como a leitura do arquivo txt, o armazenamento das coordenadas e pontos no dicionário e a função de calcular distância. A mudança ocorre na forma que ele vai encontrar a melhor rota, enquanto o algoritmo de força bruta calcula todas as possibilidades até encontrar a de menor distância e armazena-la, o vizinho mais próximo sempre procura o ponto mais próximo do atual para adiciona-lo a rota. Na etapa de resultados é analisado mais profundamente as diferenças.

Matriz pequena[4x5] com 5 pontos: 0,00099708 segundos - distância: 16 dronômetros

Matriz media[5x6] com 9 pontos: 0,00099787 segundos – distância: 20 dronômetros

Matriz grande[6x7] com 11 pontos: 0,00099849 segundos - distância: 30 dronômetros

3. Resultados

No final, ainda foi executado uma única vez uma matriz 6x7 com 12 pontos nos dois algoritmos, o algoritmo do vizinho mais próximo me retornou um tempo de execução de 0,0009980 segundos e o de força bruta me retornou 138.0820 segundos. Visto isso, nota-se que o algoritmo de força bruta é mais eficiente em relação a precisão ao calcular a menor distância, já que ele calcula todas possíveis, mas demanda muito tempo de execução quanto mais pontos existem, e o algoritmo do vizinho mais próximo é mais eficiente em relação ao tempo, em todos os testes demorou menos de 1 segundo para retornar a rota e distância percorrida, porém em quase todos eles não retornou a rota mais curta, sendo impreciso. Essa diferença vem da complexidade temporal e da heurística utilizada pelos dois algoritmos, o de força bruta possui complexidade $O(n!)$ que cresce muito mais que $O(n^2)$ do vizinho mais próximo a partir de n=4.

Desta forma, a escolha do melhor algoritmo vai depender da necessidade do usuário, se é priorizado tempo de execução o melhor é o vizinho mais próximo, se é priorizado eficiência da rota a melhor escolha é o de força bruta.

Referências Bibliográficas

Classroom PISI 2 - semana 4, “Python: Dicionários, Arquivos, Funções e módulos”, disponível em: https://classroom.google.com/c/NjI2OTgxMzYxNjg5/m/NjI2OTgxOTUyODg0/details