

Relatório - Trabalho 3 EDA II

Luis Borges 47297, Acacio Uando 55730
g318

May 26, 2024

Introdução

Este relatório descreve a implementação de um programa que ajuda Nemo e seus amigos a planejarem a rota mais eficiente em termos de consumo de energia para retornarem ao Great Barrier Reef, utilizando as correntes oceânicas a seu favor. O objetivo principal é minimizar o consumo de energia ao evitar nadar contra as correntes.

Descrição do Problema

Nemo e seus amigos enfrentam vários obstáculos e aventuras durante seu retorno para casa. Eles pretendem utilizar as correntes oceânicas para minimizar o gasto energético. O problema consiste em determinar a rota mais eficiente entre diversas localizações, levando em conta a direção e a força das correntes.

Estruturas de Dados

- **Graph:** Representa o grafo com listas de adjacência para correntes na direção favorável e listas de adjacência reversa para correntes contra a direção.

- **Location:** Classe auxiliar que armazena o índice de uma localização e a energia necessária para alcançá-la.

Algoritmo

1. **Construção do Grafo:** O grafo é construído a partir dos dados de entrada, com listas de adjacência normais e reversas.
2. **Algoritmo de Dijkstra:** Utiliza uma fila de prioridade para explorar o caminho de menor custo energético. As correntes na direção favorável têm custo zero, enquanto as correntes contra a direção têm custo igual à sua força.
3. **Processamento das Jornadas:** Para cada jornada, o algoritmo calcula o caminho de menor custo energético entre a partida e o destino.

Resultados

Para o input fornecido, o programa calcula corretamente as menores distâncias energéticas para cada jornada:

- De 2 a 1: Consumo energético de 75.
- De 4 a 3: Consumo energético de 35.
- De 3 a 5: Consumo energético de 0.

Complexidade

Complexidade Temporal

A complexidade de tempo do algoritmo de Dijkstra com fila de prioridade é $O((E + V) \log V)$, onde V é o número de vértices (localizações) e E é o número de arestas (correntes oceânicas). No pior caso, todas as arestas são processadas, e cada operação de inserção e extração na fila de prioridade leva $O(\log V)$ tempo.

- **Inicialização:** $O(V)$

- **Inserção na fila de prioridade:** $O(V \log V)$
- **Processamento de arestas:** $O(E \log V)$

Portanto, a complexidade total é $O((E + V) \log V)$.

1 Complexidade Espacial

A complexidade espacial é $O(E + V)$, devido ao armazenamento das listas de adjacência e das estruturas auxiliares (vetor de distâncias e fila de prioridade).

- **Listas de Adjacência:** $O(E + V)$
- **Vetor de Distâncias:** $O(V)$
- **Fila de Prioridade:** $O(V)$

2 Conclusão

O programa desenvolvido implementa de forma eficiente um algoritmo de Dijkstra modificado para calcular a rota de menor custo energético para Nemo e seus amigos. A escolha das estruturas de dados e a adaptação do algoritmo garantem que o problema seja resolvido dentro dos limites de tempo e espaço especificados.