BUSCA EM PROFUNDIDADE

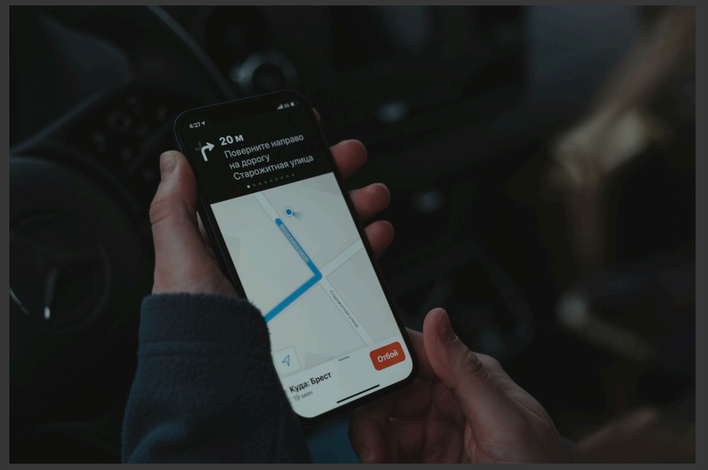
EM PARALELO

Alunos:

José Victor Rocha de Alencar
Thiago Vieira Camara

Objetivo do trabalho

Desenvolver um algoritmo que utilize a Busca em Profundidade Paralela (Parallel Depth-First Search - DFS) para explorar um grafo de fluxo na cidade. O objetivo é identificar todas as rotas alternativas entre dois pontos de interesse específicos, otimizando a busca e melhorando a eficiência no planejamento de trajetos. Esta abordagem permitirá uma análise detalhada das opções de mobilidade urbana, contribuindo para a gestão eficaz do tráfego e a melhoria da experiência dos usuários.



ALGORITMO DFS

É um algoritmo de exploração de grafos que visita vértices de forma recursiva ou iterativa. O algoritmo começa em um vértice inicial e explora o máximo possível ao longo de cada ramificação antes de retroceder.

CRIAÇÃO DO GRAFO

Foi utilizada a API do Graph hopper para gerar um grafo com os 10 pontos turísticos mais visitados de Paris e arestas conectando estes pontos.

Os pesos das arestas são o tempo necessário para ir de um ponto ao outro de bicicleta e carro.

Limitações da API impediram a criação de um grafo maior.

Navegação pelo Grafo

Foi criado um algoritmo DFS que percorre todos os caminhos possíveis entre a origem o destino, esse algoritmo então retorna o caminho mais rápido encontrado e o meio de transporte utilizado em cada parte do caminho.

Formatação do Output

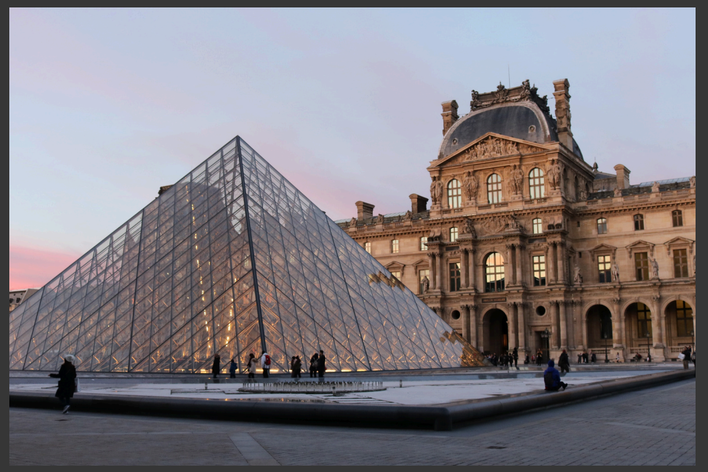
1 -carro- 8 -carro- 3

Resultados com DFS e DFS Paralelizado

- Desempenho do DFS Sequencial: O tempo de execução do algoritmo sequencial aumentou exponencialmente conforme o número de nós no grafo crescia, comportamento esperado devido à maior complexidade da exploração de todos os caminhos possíveis. Entretanto, o uso de memória manteve-se relativamente constante, uma vez que o DFS utiliza uma pilha para armazenar apenas o caminho atual.
- Otimização com DFS Paralelizado: Para melhorar o desempenho, implementamos uma versão paralelizada utilizando multithreading. O grafo foi dividido em subgrafos, e cada thread executou o DFS em um subgrafo, combinando os resultados ao final.

Análise de Complexidade

A DFS paralelizada possui uma complexidade de O(2^V), onde V é o número de vértices. Essa eficiência permite a exploração de grafos de forma ideal, seja em grafos densos ou esparsos. Com a paralelização, o trabalho pode ser distribuído entre múltiplos núcleos de processamento, reduzindo drasticamente o tempo necessário para explorar grandes grafos.



OBRIGADO PELA ATENÇÃO!!