

Relatório do trabalho prático da matéria de Teoria dos Grafos (TEG)

Guilherme M. Utiamá¹, Peter L. Brendel¹

¹Ciência da computação – Universidade do Estado de Santa Catarina (UDESC)
Joinville – SC – Brasil

utiama.guilherme@gmail.com, peter.brendel@edu.udesc.br

1. Considerações

O objetivo do trabalho é resolver as questões sugeridas pelo professor que envolvem algoritmos de busca em grafos, estudados através do material disponibilizado e as aulas teóricas. As linguagens utilizadas neste trabalho foram C++11 e JavaScript.

2. Primeira parte do trabalho

A primeira questão busca visualizar os algoritmos de busca em grafos. Dados N ($0 < N \leq 1000$) pares de coordenadas (x, y) onde $(0 \leq x, y \leq 1)$, era preciso calcular $K = \log_2(N)$ arestas que ligavam os K pontos mais próximos, de todos para todos. A primeira etapa foi feita em C++ que gera um arquivo .csv com todos os nós, suas posições e as K adjacências. Este arquivo serve para alimentar o script de JavaScript. Para executar basta abrir Index.html e fazer upload do arquivo. A seguir, foram implementadas as seguintes buscas: BFS, DFS e DFS Direcionado. A direção do DFS seguia a direção do eixo X, i.e, se $A.x < B.x$ então $A \rightarrow B$ senão $B \rightarrow A$. No BFS, era preciso mostrar a profundidade de cada nó. Isso foi feito colorindo-os conforme eram encontrados na busca, seguindo uma lógica onde era preciso encontrar M filhos antes da profundidade avançar.

3. Segunda e terceira parte (Dijkstra modificado)

A modificação do dijkstra ocorre da seguinte maneira: o principal foco destas questões é não calcular todas as rotas possíveis a partir de um nó, e sim quando se chegar no fim, parar a função de dijkstra para aquele caminho e percorrer para o próximo (caso exista!). Então, para isso, adicionou-se uma função de break toda vez que chegasse no nó de chegada e percorria para o próximo caminho. Com isso, na questão 2 foi mostrado na tela o menor custo do caminho a partir de um nó até outro. E na questão 3 se utilizou a mesma função, porém, a diferença é que não se utilizou apenas um único nó de origem, foram utilizados todos os nós no dijkstra, mas todos eles com a mesma chegada. Assim, o programa irá mostrar o custo mínimo de todos os nós até um de chegada.