
Relatório

Atividade 1

UNIVERSIDADE FEDERAL DE SANTA CATARINA

25 DE JULHO DE 2021

SAMUEL CARDOSO
19100544

JOSÉ LUIZ DE SOUZA
19100533

Questão 1

O grafo foi feito com dicionários. Cada vertice mantém um dicionário que é uma lista de vizinhos, no qual a chave é o índice do vizinho e o valor armazenado é o peso da aresta. Com isso algumas funções acabam sendo instantaneas como saber se o vertice 1 é vizinho do vertice 2.

Nos vertices foi utilizado um vetor extra para manter o controle dos vizinhos pois algumas funções acessam a lista de vizinhos de um vertice, no qual iria ser demorado se utilizasse apenas um dicionário.

Questão 2

No bfs foi utilizado uma estrutura do python chamada deque no qual permite o acesso com complexidade $O(1)$ na inserção e remoção de elementos da fila, que era o que estavamos buscando. A estrutura normal do python utiliza complexidade $O(n)$ para remoção. Além disso foi utilizado vetores para manter o controle de quais vertices foram acessados.

Questão 3

No ciclo euleriano utilizamos um dicionário para armazenar se uma aresta foi visitada. Utilizamos uma tupla como chave do dicionário no qual os valores da tupla são os índices dos vertices ligados. Com isso o acesso ao percorrer a aresta podemos alterar o valor em $O(1)$.

Questão 4

A implementação de dijkstra foi bem simples, basicamente uma transposição do algoritmo encontrado nas notas de aula para python. Onde então utilizamos de 3 listas para armazenar os pesos, os antecedentes e o estado do vértice (se ele já foi acessado ou não). Também foi utilizada a nossa lista encadeada para a procura dos vértices vizinhos.

Questão 5

Como nossa estrutura utiliza listas encadeadas, a função de warshall acaba sendo comprometida já que utiliza normalmente matizes, com isso perdemos um pouco de velocidade na execução desse algoritmo, já que para sua execução foi realizada uma simulação de matrizes através das listas encadeadas.