



UNIVERSIDADE FEDERAL DE SANTA CATARINA

CAMPUS Trindade

DEPARTAMENTO INE

CURSO Ciências da Computação

Carlos Eduardo Vitorino Gomes (23150560)

Eduardo Cunha Cabral (23150561)

Enrico Caliollo (23150562)

Grafos: AT2

Introdução e Dificuldades Encontradas

Fizemos nosso trabalho utilizando a IDE VSCode e Python em diferentes versões, dentre elas, 3.12.3 e 3.6.15. Localmente, em diferentes computadores e sistemas operacionais, todos os arquivos rodam corretamente, porém, ao testar esses mesmos arquivos no Moodle (alterando apenas o método com que a entrada é recebida), o desafio de número 2 apresentou problemas, quando o teste referente ao arquivo manha.net é executado, o resultado não é o esperado. Como um grafo pode ter diferentes ordenações topológicas, talvez o resultado que o Moodle espera seja diferente do resultado do nosso código, ainda que ambos estejam certos.

1) COMPONENTES FORTEMENTE CONEXAS

Para a busca em largura, usamos uma hash table para manter as informações específicas de cada vértice, como distância, se o vértice é conhecido, e seu f. Utilizamos essa estrutura, pois é mais confiável usar o index do vértice como o valor de hash do que usar o index de uma lista.

Usamos também uma classe para armazenar a variável `tempo` pois, durante as recursões, o valor dela era zerado. Dessa forma, temos um tempo `global`.

2) ORDENAÇÃO TOPOLÓGICA

Para a busca em largura, usamos uma hash table para manter as informações específicas de cada vértice, como distância, se o vértice é conhecido, e seu f. Utilizamos essa estrutura, pois é mais confiável usar o index do vértice como o valor de hash do que usar o index de uma lista.

Usamos também uma classe para armazenar a variável `tempo` pois, durante as recursões, o valor dela era zerado. Dessa forma, temos um tempo `global`.

3) ÁRVORES GERADORAS MÍNIMAS

Escolhemos o algoritmo Kruskal para achar a árvore geradora mínima do grafo, por ser mais simples e mais eficiente. Usamos uma lista para armazenar a árvore e um hash table de sets para armazenarmos a árvore de cada vértice. Usamos sets por ter como diferença a unicidade de elementos e facilidade para fazermos a união de sets diferentes.