TRABALHO DE GRUPO 2 (T2) DE DESENHO DE ALGORITMOS 21/22:

AGÊNCIA

DE

VIAGENS

Grupo G58:

- Gabriel Augusto Rocco (up201800172)
- Guilherme Miguel de Lima Freire (up202004809)
- Rafael Azevedo Alves (up202004476)

Descrição do problema

Neste trabalho, pretende-se implementar um sistema capaz apoiar a gestão de pedidos para transporte de grupos de pessoas de um local de origem para um local de destino.

Para a criação de algoritmos que permitam um desenvolvimento desse sistema, são nos transmitidas várias informações sobre o modo de funcionamento dos transportes:

- A empresa dispõe de **veículos em vários locais**;
- Cada um fará um **único trajeto** de uma origem para um destino;
- Cada trajeto tem uma **certa capacidade**;
- O transporte demora um **certo tempo** (dado em horas) e tem um **certo custo de transporte** (bilhete) **por pessoa**.

Descrição da solução – Cenário 1

Formalização

Dados de entrada:

- T = conjunto de transportes:
  - o_t – origem do transporte;
  - ds_t – destino do transporte;
  - c_t – capacidade do transporte;

Variáveis de decisão:

- MAX c_t dentro de um encaminhamento entre o_t e ds_t;
- Np = número de pessoas que se vai transportar;

Objetivo:

- Maximizar dimensão do grupo;

Sujeito a:

- Np <= MAX c_t dentro de um encaminhamento entre o_t e ds_t

Descrição da solução – Cenário 1

Descrição de Algoritmos Relevantes

O **Cenário 1**, no qual os grupos não se separam:

- Na alínea 1.1, o objetivo é maximizar a dimensão do grupo e indicar um qualquer encaminhamento.

Para a resolução deste problema foi aplicado um algoritmo de **Widest Path**.

- Na alínea 1.2, o objetivo é maximizar a dimensão do grupo e minimizar o número de transbordos, sem privilegiar um dos critérios relativamente ao outro, apresentando alternativas pareto-ótimas, se existirem. Tal significa que um grupo maior pode ser transportado se se admitir mais transbordos; ou se se quiser ter menos transbordos, a dimensão do grupo pode ter de ser menor.

Para a resolução deste problema foi aplicado um algoritmo **Breadth-first search** (**BFS**).

Descrição da solução – Cenário 1

Análise de Complexidade

Complexidade Temporal

- Na alínea 1.1 é $O(|E|\log|V|)$
- Na alínea 1.2 é $O(|V| + |E|)$

Complexidade Espacial

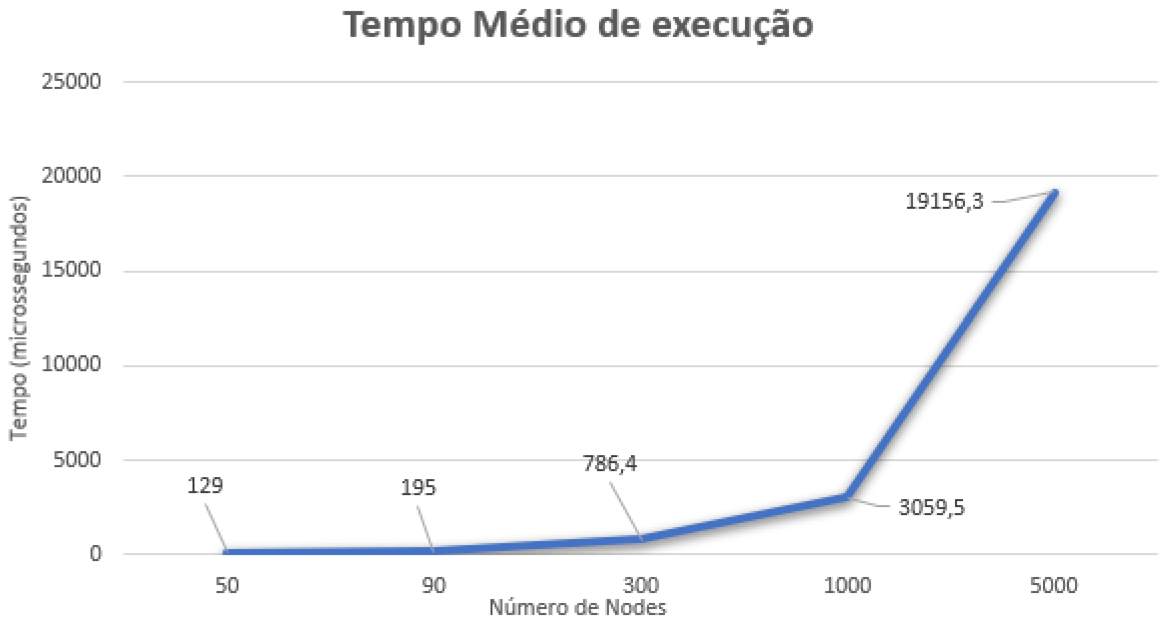
- $O(|V|)$

Descrição da solução – Cenário 1

Avaliação Empírica

Foram realizados testes com os ficheiros de 50, 90, 300, 1000 e 5000 nodes que foram dados com o enunciado do trabalho com a dimensão de grupo de 20 pessoas, e foram analisados os tempos de execução que se encontram nos seguintes gráficos.

Alínea 1.1



Alínea 1.2

Para o ficheiro de 50 nodes demorou 13894941 microssegundos e para os restantes demorou um tempo bastante elevado para executar, logo não estão apresentados os valores.

Descrição da solução – Cenário 2

Formalização

Dados de entrada:

- T = conjunto de transportes:
  - o_t – origem do transporte;
  - ds_t – destino do transporte;
  - c_t – capacidade do transporte;
  - d_t – duração do transporte;

Variáveis de decisão:

- MAX_FLOW = quantidade de pessoas que já se transportou;
- Np = número de pessoas que faltam transportar;
- MAX_ENC = capacidade máxima de cada encaminhamento;
- Dg = número do grupo de pessoas que se vai transportar;

Objetivo:

- Maximizar dimensão do grupo procurando caminhos com MAX_ENC ;
- Minimizar Np;

Sujeito a:

Dimensão do grupo <= MAX_FLOW + Bottle Neck

Bottle Neck corresponde às pessoas que se pode mandar por um caminho que se encontrou

Descrição da solução – Cenário 2

Descrição de Algoritmos Relevantes

O **Cenário 2**, no qual os grupos se podem separam:

- Na alínea 2.1, o objetivo é determinar um encaminhamento para um grupo, dada a sua dimensão.
- Na alínea 2.2, o objetivo é corrigir um encaminhamento, se necessário, para que a dimensão do grupo possa aumentar de um número de unidades dado.
- Na alínea 2.3, o objetivo é determinar a dimensão máxima do grupo e um encaminhamento.

Para a resolução destes três problemas foi aplicado o **Algoritmo de Edmonds-Karp**.

- Na alínea 2.4, o objetivo é Partindo de um encaminhamento que constitui um grafo acíclico, determinar quando é que o grupo se reuniria novamente no destino, no mínimo.
- Na alínea 2.5, nas condições anteriores (alínea 2.4), admitindo que os elementos que saem de um mesmo local partem desse local à mesma hora (e o mais cedo possível), indicar o tempo máximo de espera e os locais em que haveria elementos que esperam esse tempo.

Para a resolução destes dois problemas foi aplicado um algoritmo **Critical Path Method (CPM)**.

Descrição da solução – Cenário 2

Análise de Complexidade

Complexidade Temporal

- Nas alíneas 2.1, 2.2 e 2.3 é O(|V|*|E|^2)
- Nas alíneas 2.4 e 2.5 é O(|V|+|E|)

Complexidade Espacial

- O(|V|)

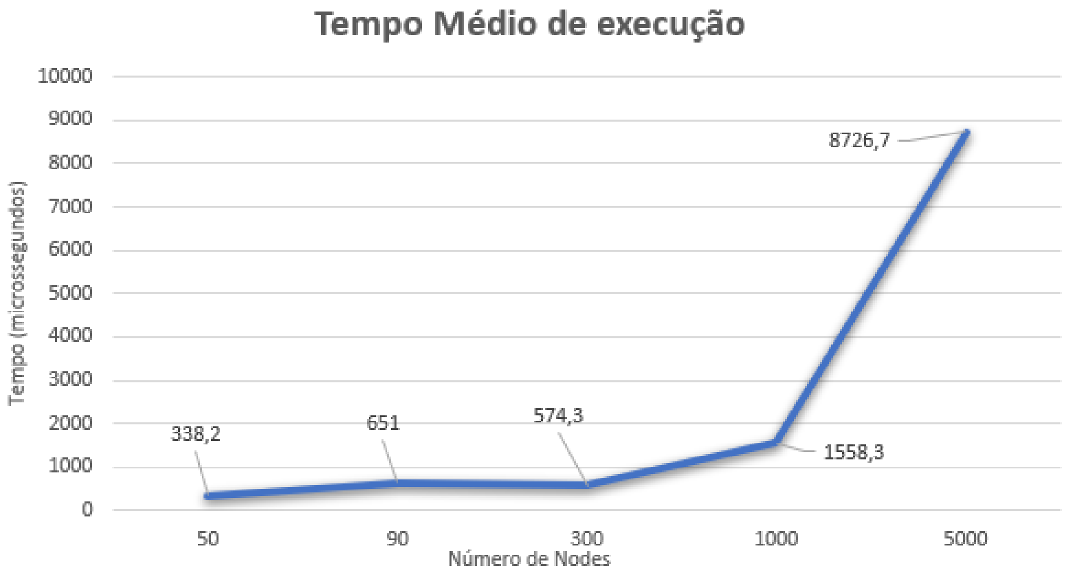
Descrição da solução – Cenário 2

Avaliação Empírica

Foram realizados testes com os ficheiros de 50, 90, 300, 1000 e 5000 nodes que foram dados com o enunciado do trabalho com a dimensão de grupo de 20 pessoas, e foram analisados os tempos de execução que se encontram nos seguintes gráficos.

Alíneas 2.1, 2.2 e 2.3

Alíneas 2.4 e 2.5



Principais dificuldades & Esforço de cada elemento

O grupo conseguiu executar todas as tarefas propostas para o trabalho em questão.

A principal dificuldade foi perceber qual o algoritmo que melhor se adequa aos diferentes cenários que tivemos de implementar para tentar obter a melhor solução possível para cada um.

Apesar das dificuldades encontradas podemos afirmar que após a realização do trabalho conseguimos perceber melhor cada um dos algoritmos estudados na disciplina de Desenho de Algoritmos, assim como a utilização dos diferentes tipos de algoritmos, e em que situações melhor se adequa o uso dos mesmos para melhorar o resultado que se pretende atingir. Conseguimos, também, melhorar o nosso conhecimento sobre a manipulação de grafos.

O trabalho foi dividido de maneira igual por todos os elementos do grupo e todos contribuíram para o sucesso do mesmo.

- Gabriel Rocco: 33.3%
- Guilherme Freire: 33.3%
- Rafael Alves: 33.3%