Relatório 2º projecto ASA 2019/2020

Grupo: al026

Aluno(s): António Romeu Pinheiro (92427) e Mariana Cintrão (93737)

Descrição do Problema e da Solução

O problema descrito no enunciado trata o algoritmo que permite calcular o número máximo de clientes que podem ir ao supermercado tendo em conta o distanciamento social, que no caso consiste em que os clientes não utilizem as mesmas ruas e cruzamentos para se deslocarem até aos supermercados.

A nossa solução passa por aplicar ao problema um algoritmo de fluxo máximo (*Ford-Fulkerson*) a um grafo no qual cada cruzamento é representado por dois vértices (um de entrada e outro de saída de fluxo) unidos por uma aresta. Desta forma, pode-se verificar que apenas um cliente passa por cada cruzamento, sendo esta passagem representada pelo fluxo em si. Um cruzamento que esteja ligado a outro por uma rua faz parte da lista de adjacências do segundo (e vice-versa). Cada aresta tem no máximo capacidade 1. Para todas as arestas existe uma aresta residual com direção oposta, cuja capacidade é inicializada a 0.

Cada cliente e supermercado corresponde a um cruzamento. A source (s) encontra-se ligada a cada cliente e cada supermercado encontra-se ligado ao sink (t), de maneira que o fluxo total seja incrementado sempre que exista um caminho entre s e t.

Análise Teórica

V é dado por *número de avenidas* * *número de ruas* recebidos no input. O número de vértices criados no grafo é 2V + 2, contemplando uma source e um sink.

- Leitura do input: O(V);
- Inicialização do vetor de pais (π): O(V) - std::vector::push_back tem complexidade de O(1) e é executado 2V + 2 vezes;
- Construção do vetor de adjacências na criação do grafo: O(V) - nesta etapa, a cada vértice podem ser adicionadas no máximo 5 adjacências e std::unordered_map::insert de um só elemento tem complexidade máxima linear, dependendo da quantidade de elementos;
- Adição das adjacências para e dos vértices que representam mercados: O(V) - no máximo existem 2V + 2 mercados e a inserção de adjacências é feita num mapa cujo tamanho máximo é 5;

Relatório 2º projecto ASA 2019/2020

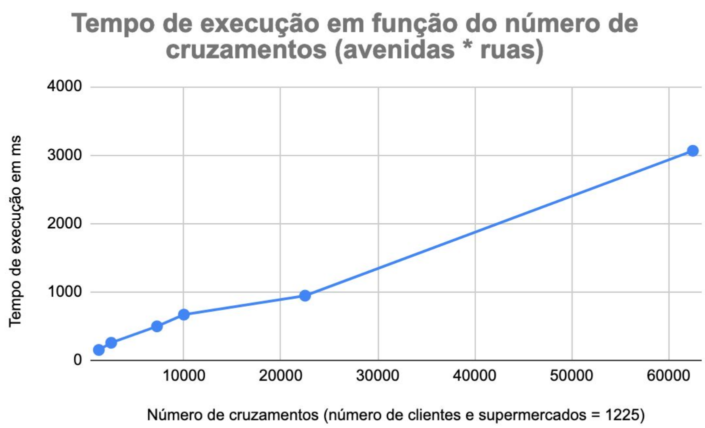
Grupo: al026

Aluno(s): António Romeu Pinheiro (92427) e Mariana Cintrão (93737)

- Adição das adjacências para e dos vértices que representam clientes: O(V) - no máximo existem 2V + 2 clientes e a inserção de adjacências é feita num mapa cujo tamanho máximo é 6;
- Aplicação do algoritmo *Ford-Fulkerson* no grafo para calcular o fluxo máximo de s para t: O(V^2) - O(E * f) em que E é o número máximo de arestas E = 7 * (2V + 2) - 7 representa o número máximo de adjacência que cada vértice pode ter - e em que f (fluxo máximo) é igual a V. O(V * (14V + 14)) = O(V^2);
- Apresentação dos dados: O(1).

Complexidade global da solução: O(V^2).

Avaliação Experimental dos Resultados



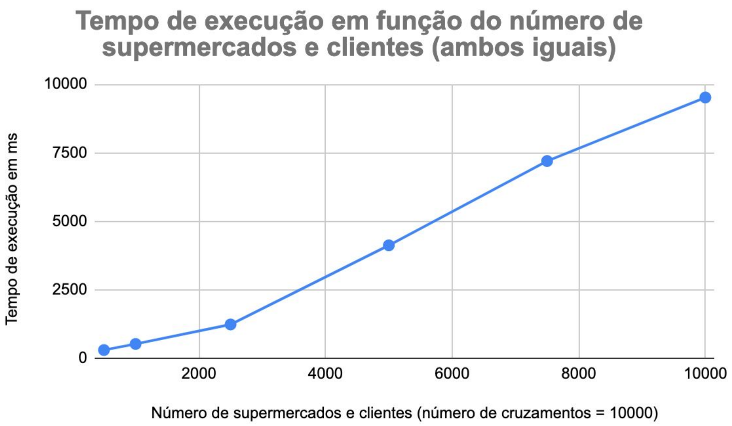
1ª experiência: com um número constante e igual de supermercados e clientes (1225 cada) e variando apenas o número de cruzamentos (quantidade de avenidas igual à quantidade de ruas) é de notar que a complexidade temporal é linear como seria expectável.

A complexidade do algoritmo *Ford-Fulkerson* - O(Ef) - (em que E corresponde a 7 * (2V + 2) e f é constante pois depende maioritariamente do número de supermercados e clientes) manifesta-se na forma O(V).

Relatório 2º projecto ASA 2019/2020

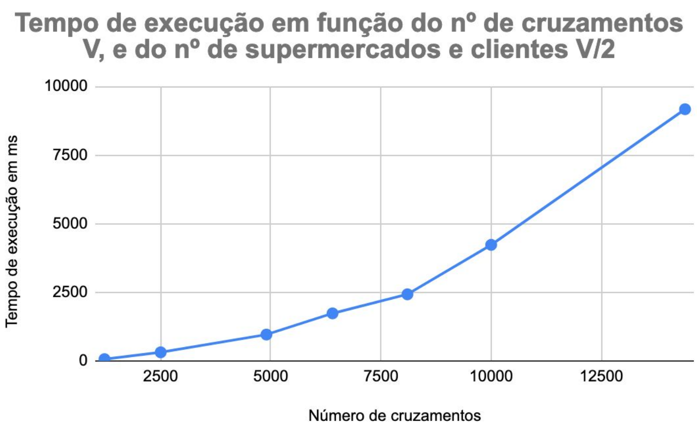
Grupo: al026

Aluno(s): António Romeu Pinheiro (92427) e Mariana Cintrão (93737)



2ª experiência: o grafo gerado tem um número de cruzamentos constante (10000) e o número de supermercados e clientes (ambos com o mesmo valor) é incrementado ao longo dos testes.

A complexidade desta experiência é O(V) uma vez que, de acordo com a complexidade de *Ford-Fulkerson* - O(Ef) - como o número de arestas se mantém constante (E = 10000) e o fluxo varia com o aumento do número de supermercados e clientes, podemos concluir que o tempo aumenta de forma linear.



3ª experiência: por fim, quando variamos tanto o número de cruzamentos V como o número de supermercados e clientes V/2, evidencia-se a natureza quadrática do algoritmo. O fluxo máximo passa a ser V/2 e a quantidade máxima de arestas é mais uma vez 7 * (2V + 2) o que faz da complexidade temporal do algoritmo O(V^2).

De acordo com a nossa análise teórica, os resultados obtidos através das diversas experiências estão de acordo com os resultados esperados uma vez que, através dos gráficos podemos verificar que um aumento, tanto do número de avenidas e ruas, como do número de supermercados e clientes, leva a um aumento quadrático do tempo de execução.