



Projeto e Análise de Algoritmos
Trabalho 4º Bimestre
Algoritmos para Grafos e Redes de Fluxo

O trabalho está dividido em várias propostas. Cada grupo, através de sorteio, ficou responsável por resolver uma proposta das listadas a seguir.

O QUE DEVE SER ENTREGUE

Relatório em formato pdf contendo:

- Descrição de como foi realizado o processo empírico de determinação dos custos: cenário de realização dos experimentos e como foram tomadas as métricas exigidas.
 - Detalhar a configuração usada nos testes (Processador, SO, IDE etc.).
- Gráficos evidenciando o comportamento dos métodos perante todos os cenários considerando o tamanho dos conjuntos de entrada.
- Análise do comportamento dos métodos durante a execução dos testes.
 - Esta análise deve ser feita com bastante critério e ser esclarecedora, apontando razões para os comportamentos observados.
- O formato dos relatórios deve seguir o padrão da SBC conforme disponível no endereço: <https://www.sbc.org.br/documentos-da-sbc/summary/169-templates-para-artigos-e-capitulos-de-livros/878-modelosparapublicaodeartigos>



Códigos fonte usados na execução dos experimentos

- Não é necessário o envio dos projetos, apenas os códigos fonte;
- Manter no código os comandos usados para tomada de tempo e contagem do número de comparações (quanto pertinente).

Quando:

A data de entrega está marcada para dia **04/08/2022 até as 23:59**.

A entrega do trabalho deve ser feita via Teams.

Para este trabalho não haverá a etapa de apresentação.

PROPOSTA 1 – Pedro Xavier e Vitor Gilnek

Tarefa:

- a) Implementar o algoritmo de Dijkstra de forma que todos os vértices sejam o ponto de origem. Ou seja, o método será executado V vezes, uma para cada vértice de origem.
- b) Implementar o método de Floyd-Warshall

Como:

- A linguagem utilizada no desenvolvimento é de vossa escolha.
- A forma com que os métodos serão implementados é determinada pelo grupo. No entanto, deve-se adotar a mesma abordagem para ambos os métodos. Por exemplo, ambos devem ser iterativos ou ambos recursivos. Ambos devem trabalhar sobre a mesma representação (lista ou matriz de adjacência).
- Os dados a serem empregados na construção dos grafos estão disponíveis no endereço a seguir:
<https://drive.google.com/file/d/1e9F7pIPWXMxQDrWcsFBX0YYLDz2m0rwK/view?usp=sharing>

Critérios que devem ser analisados:

- a) Tempo cronológico para a execução do algoritmo de Floyd-Warshall;
- b) Tempo cronológico para V execuções do algoritmo de Dijkstra;
- c) Número de testes para checar se o novo caminho é melhor ou pior que o já conhecido (não precisa considerar os testes para verificar se o vértice já foi processado ou não)

PROPOSTA 2 – Ana Paula Bernardes e Willian Faino

Tarefa:

- a) Implementar o algoritmo de Dijkstra usando uma lista para gerenciar os vértices e suas distâncias. Como as distâncias vão sendo atualizadas conforme os vértices vão sendo descobertos não há como ordenar a lista desde o princípio.
- b) Implementar o algoritmo de Dijkstra usando um heap de mínimo para gerenciar os vértices e suas distâncias. Nesse caso o vértice mais próximo sempre estará no topo da estrutura.

Como:

- A linguagem utilizada no desenvolvimento é de vossa escolha.
- A forma com que os métodos serão implementados é determinada pelo grupo. No entanto, deve-se adotar a mesma abordagem para ambos os métodos. Por exemplo, ambos devem ser iterativos ou ambos recursivos. Ambos devem trabalhar sobre a mesma representação (lista ou matriz de adjacência).
- Considerem como vértice inicial sempre o primeiro vértice do conjunto.
- Os dados a serem empregados na construção dos grafos estão disponíveis no endereço a seguir:
<https://drive.google.com/file/d/1e9F7pIPWXMxQDrWcsFBX0YYLDz2m0rwK/view?usp=sharing>

Critérios que devem ser analisados:

- a) Tempo cronológico para a execução do algoritmo usando listas;
- b) Tempo cronológico para a execução do algoritmo usando o heap de mínimo;
- c) Número de comparações usadas para construir o heap e mantê-lo ordenado ao longo das execuções;
- d) Número de comparações necessárias para encontrar o menor elemento a cada iteração do algoritmo. Nessa conta devem entrar também as comparações necessárias para encontrar o vértice para atualizar seu custo quando um novo caminho é encontrado (relaxamento).

PROPOSTA 3 - Igor Engler e Marcos Mucelini

Tarefa:

- a) Implementar o algoritmo de Dijkstra usando uma matriz de adjacências como forma de representação dos vértices e arestas.
- b) Implementar o algoritmo de Dijkstra usando uma lista de adjacência como estratégia de representação dos elementos.

Como:

- A linguagem utilizada no desenvolvimento é de vossa escolha.
- A forma com que os métodos serão implementados é determinada pelo grupo. No entanto, deve-se adotar a mesma abordagem para ambos os métodos. Por exemplo, ambos devem ser iterativos ou ambos recursivos. Ambos devem trabalhar sobre a mesma representação (lista ou matriz de adjacência).
- Considerem como vértice inicial sempre o primeiro vértice do conjunto.
- Os dados a serem empregados na construção dos grafos estão disponíveis no endereço a seguir:
<https://drive.google.com/file/d/1e9F7pIPWXMxQDrWcsFBX0YYLDz2m0rwK/view?usp=sharing>

Critérios que devem ser analisados:

- a) Tempo cronológico para a execução do algoritmo usando matriz de adjacências;
- b) Tempo cronológico para a execução do algoritmo usando a lista de adjacências;
- c) Número de comparações usadas para pegar o valor e testar os pesos de cada aresta;

PROPOSTA 4 – João Dota, Rodrigo Campos e Valquíria Belusso

Tarefa:

- a) Implementar o algoritmo Jonhson para encontrar as menores distâncias entre todos os pares de vértices.
- b) Implementar o algoritmo Floyd-Warshall para encontrar as menores distâncias entre todos os pares de vértices

Como:

- A linguagem utilizada no desenvolvimento é de vossa escolha.
- A forma com que os métodos serão implementados é determinada pelo grupo. No entanto, deve-se adotar a mesma abordagem para ambos os métodos. Por exemplo, ambos devem ser iterativos ou ambos recursivos. Ambos devem trabalhar sobre a mesma representação (lista ou matriz de adjacência).
- Os dados a serem empregados na construção dos grafos estão disponíveis no endereço a seguir:
<https://drive.google.com/file/d/1e9F7pIPWXMxQDrWcsFBX0YYLDz2m0rwK/view?usp=sharing>

Critérios que devem ser analisados:

- a) Tempo cronológico para a execução do algoritmo de Floyd-Warshall;
- b) Tempo cronológico para a execução do algoritmo de Johnson;
- c) Número de testes para checar se o novo caminho é melhor ou pior que o já conhecido (não precisa considerar os testes para verificar se o vértice já foi processado ou não)

PROPOSTA 5 – Leonardo Vanzin, Lucas Veit e Roberta Aparecida Alcantara

Tarefa:

- a) Implementar o algoritmo de fluxo máximo de Ford-Fulkerson para encontrar o valor da maior quantidade de fluxo que pode ser passada pela rede fornecida.
- b) Implementar o algoritmo de fluxo máximo de Edmonds-Karp para encontrar o valor da maior quantidade de fluxo que pode ser passada pela rede fornecida.

Como:

- A linguagem utilizada no desenvolvimento é de vossa escolha.
- O vértice inicial (fonte) para o método será sempre o primeiro vértice presente no arquivo de entrada.
- O vértice final (escoadouro) para o método será sempre o último vértice presente no arquivo de entrada.
- A forma com que os métodos serão implementados é determinada pelo grupo. No entanto, deve-se adotar a mesma abordagem para ambos os métodos. Por exemplo, ambos devem ser iterativos ou ambos recursivos. Ambos devem trabalhar sobre a mesma representação (lista ou matriz de adjacência).
- Os dados a serem empregados na construção dos grafos estão disponíveis no endereço a seguir:
<https://drive.google.com/file/d/1pY47YMRjpqNRrx6GNJpRAgiPusw1hWF4/view?usp=sharing>

Critérios que devem ser analisados:

- a) Tempo cronológico para a execução do algoritmo de Ford-Fulkerson;
- b) Tempo cronológico para a execução do algoritmo de Edmonds-Karp;
- c) Comparativo entre os fluxos encontrados pelas duas estratégias.
- d) Número de caminhos com capacidade residual diferente de zero que foram processados.

Tarefa:

- a) Implementar o algoritmo de Prim para encontrar a árvore geradora mínima a partir do grafo de entrada.
- b) Implementar o algoritmo Kruskal para encontrar a árvore geradora mínima a partir do grafo de entrada.
- c)

Como:

- A linguagem utilizada no desenvolvimento é de vossa escolha.
- O vértice inicial para o método de Prim pode ser definido como o primeiro vértice presente no arquivo de entrada.
- A forma com que os métodos serão implementados é determinada pelo grupo. No entanto, deve-se adotar a mesma abordagem para ambos os métodos. Por exemplo, ambos devem ser iterativos ou ambos recursivos. Ambos devem trabalhar sobre a mesma representação (lista ou matriz de adjacência).
- Os dados a serem empregados na construção dos grafos estão disponíveis no endereço a seguir:
<https://drive.google.com/file/d/1e9F7pIPWXMxQDrWcsFBX0YYLDz2m0rwK/view?usp=sharing>

Critérios que devem ser analisados:

- e) Tempo cronológico para a execução do algoritmo de Prim;
- f) Tempo cronológico para a execução do algoritmo de Kruskal;
- g) Número de testes para decidir qual será a próxima aresta a ser escolhida ou o número de comparações entre os pesos das arestas para fazer processos de ordenação (caso seja adotada tal estratégia).

PROPOSTA 7 – Angelo Orssatto e Gabriel Girardi

Tarefa:

- a) Implementar o algoritmo de Dijkstra para encontrar as menores distâncias a partir de uma origem única.
- b) Implementar o algoritmo de Bellman-Ford para encontrar as menores distâncias a partir de uma origem única.

Como:

- A linguagem utilizada no desenvolvimento é de vossa escolha.
- Para os dois métodos o vértice inicial deve ser o mesmo. Para tanto, pode-se definir o ponto de partida como sendo o primeiro vértice presente no grafo.
- A forma com que os métodos serão implementados é determinada pelo grupo. No entanto, deve-se adotar a mesma abordagem para ambos os métodos. Por exemplo, ambos devem ser iterativos ou ambos recursivos. Ambos devem trabalhar sobre a mesma representação (lista ou matriz de adjacência).
- Os dados a serem empregados na construção dos grafos estão disponíveis no endereço a seguir:
<https://drive.google.com/file/d/1e9F7pIPWXMxQDrWcsFBX0YYLDz2m0rwK/view?usp=sharing>

Critérios que devem ser analisados:

- a) Tempo cronológico para a execução do algoritmo de Dijkstra;
- b) Tempo cronológico para a execução do algoritmo de Bellman-Ford;
- c) Número de testes para checar se o novo caminho é melhor ou pior que o já conhecido (não precisa considerar os testes para verificar se o vértice já foi processado ou não)

Proposta 8 – Alonso Fritz e Maurício Ishida

Tarefa:

- a) Implementar o algoritmo de Busca em Profundidade sobre um grafo não direcionado.
- b) Implementar o algoritmo de Busca em Largura para percorrer um grafo não direcionado.

Como:

- A linguagem utilizada no desenvolvimento é de vossa escolha.
- No momento de executar os métodos considerem que eles percorrerão todos os vértices, simulando assim uma busca sem sucesso sobre o grafo.
- Para os dois métodos o vértice inicial deve ser o mesmo. Para tanto, pode-se definir o ponto de partida como sendo o primeiro vértice presente no grafo.
- A forma com que os métodos serão implementados é determinada pelo grupo. No entanto, deve-se adotar a mesma abordagem para ambos os métodos. Por exemplo, ambos devem ser iterativos ou ambos recursivos. Ambos devem trabalhar sobre a mesma representação (lista ou matriz de adjacência).
- Os dados a serem empregados na construção dos grafos estão disponíveis no endereço a seguir:
<https://drive.google.com/file/d/1e9F7pIPWXMxQDrWcsFBX0YYLDz2m0rwK/view?usp=sharing>

Critérios que devem ser analisados:

- a) Tempo cronológico para a execução do algoritmo de Busca em Largura;
- b) Tempo cronológico para a execução do algoritmo de Busca em Profundidade;
- c) Tamanho máximo alcançado pela pilha ou fila ao longo da execução de cada grafo.

Tarefa:

- a) Implementar o algoritmo de fluxo máximo de Ford-Fulkerson para encontrar o valor da maior quantidade de fluxo que pode ser passada pela rede fornecida.
- b) Implementar o algoritmo de fluxo máximo de Push-Relabel para encontrar o valor da maior quantidade de fluxo que pode ser passada pela rede fornecida.

Como:

- A linguagem utilizada no desenvolvimento é de vossa escolha.
- O vértice inicial (fonte) para o método será sempre o primeiro vértice presente no arquivo de entrada.
- O vértice final (escoadouro) para o método será sempre o último vértice presente no arquivo de entrada.
- A forma com que os métodos serão implementados é determinada pelo grupo. No entanto, deve-se adotar a mesma abordagem para ambos os métodos. Por exemplo, ambos devem ser iterativos ou ambos recursivos. Ambos devem trabalhar sobre a mesma representação (lista ou matriz de adjacência).
- Os dados a serem empregados na construção dos grafos estão disponíveis no endereço a seguir:
<https://drive.google.com/file/d/1pY47YMRjpqNRrx6GNJpRAgiPusw1hWF4/view?usp=sharing>

Critérios que devem ser analisados:

- a) Tempo cronológico para a execução do algoritmo de Ford-Fulkerson;
- b) Tempo cronológico para a execução do algoritmo de Push-Relabel;
- c) Comparativo entre os fluxos encontrados pelas duas estratégias.
- d) Número de caminhos com capacidade residual diferente de zero que foram processados pelo Ford-Fulkerson.
- e) Número médio de vezes que cada vértice foi processado no método de Push-Relabel.

Tarefa:

- a) Implementar o algoritmo Kruskal para encontrar a árvore geradora mínima (AGM) a partir do grafo de entrada usando como estratégia de gerenciamento dos conjuntos uma abordagem quadrática.
- b) Implementar o algoritmo Kruskal para encontrar a árvore geradora mínima a partir do grafo de entrada usando como estratégia de gerenciamento dos conjuntos a abordagem Union-find (hierárquica) cuja complexidade assintótica é $O(E \log V)$.

Como:

- A linguagem utilizada no desenvolvimento é de vossa escolha.
- A forma com que os métodos serão implementados é determinada pelo grupo. No entanto, deve-se adotar a mesma abordagem para ambos os métodos. Por exemplo, ambos devem trabalhar sobre a mesma representação (lista ou matriz de adjacência).
- Os dados a serem empregados na construção dos grafos estão disponíveis no endereço a seguir:
<https://drive.google.com/file/d/1e9F7pIPWXMxQDrWcsFBX0YYLDz2m0rwK/view?usp=sharing>

Critérios que devem ser analisados:

- a) Tempo cronológico para a execução do algoritmo de Kruskal quadrático;
- b) Tempo cronológico para a execução do algoritmo de Kruskal baseado em union-find;
- c) Número de testes para descobrir se a aresta escolhida pode ser adicionada à AGM ou se ela fecha um ciclo.