

Relatório 2º projecto ASA 2023/2024

Grupo: TP062

Aluno: Francisco Ferro Pereira (107502)

Descrição do Problema e da Solução

A solução proposta envolve a procura de componentes fortemente ligados (SCCs) no grafo especificado no input e posteriormente uma travessia DFS a começar em todos os nós do grafo, a fim de encontrar o caminho mais longo possível num grafo dirigido acíclico (DAG). O valor obtido corresponde ao máximo número de saltos que a doença pode fazer no grafo em questão. Para a representação do grafo foi utilizada uma matriz de adjacências. Para a procura de componentes fortemente ligados é utilizado o algoritmo Kosaraju-Sharir que envolve duas travessias DFS, uma no grafo original e outra no grafo transposto. Caso sejam detectados componentes fortemente ligados no grafo de input, os mesmos são colapsados em um nó, por forma a transformar o grafo de input em um DAG. De seguida, itera-se por todos os vértices do grafo e é feita a travessia DFS a começar em cada um dos vértices. Durante a execução destes DFS, utilizamos uma abordagem de programação dinâmica com memoization, guardando as soluções dos subproblemas por forma a evitar recomputações desnecessárias.

Análise Teórica

- Leitura de input:

- $O(1)$ para o número de nós e número de arestas do grafo.
- A leitura das arestas do grafo depende do número das mesmas (E). Esta tarefa é portanto linear com o número de arestas, por isso $O(E)$.
- A matriz de adjacências que representa o grafo tem como dimensões $V \times V$, sendo V o número de vértices do grafo. A inicialização desta estrutura cresce quadraticamente com V , logo esta operação custa $O(V^2)$, sendo por isso o fator dominante na leitura de input.

Em suma, a complexidade temporal da leitura de input é $O(V^2)$.

- Algoritmo Kosaraju-Sharir:

- Inicialização de vetor de vértices visitados e vetor de cores dos mesmos é linear com o número de vértices, logo, $O(V)$.
- Para cada vértice no grafo é feita uma primeira travessia DFS. A complexidade da DFS é $O(V)$, pois a procura de vizinhos implica iterar por todos os vértices. Esta tarefa custa portanto $O(V^2)$.
- A transposição do grafo custa $O(V^2)$.
- A segunda travessia DFS também custa $O(V^2)$.
- Colapsar um SCC em um nó custa $O(V \cdot S)$, em que S é o número de nós no SCC. No pior caso o grafo todo é um SCC, logo colapsar custa $O(V^2)$.

Em suma, a complexidade temporal do algoritmo Kosaraju-Sharir é $O(V^2)$.

- DFS para caminho mais longo:

- Encontrar os nós do grafo depois de colapsar os SCCs custa $O(V^2)$.
- Para cada vértice é feita uma travessia DFS que custa $O(V)$. Logo esta tarefa custa $O(V^2)$.

Em suma, a complexidade temporal do algoritmo para encontrar o caminho mais longo num DAG é $O(V^2)$.

Relatório 2º projecto ASA 2023/2024

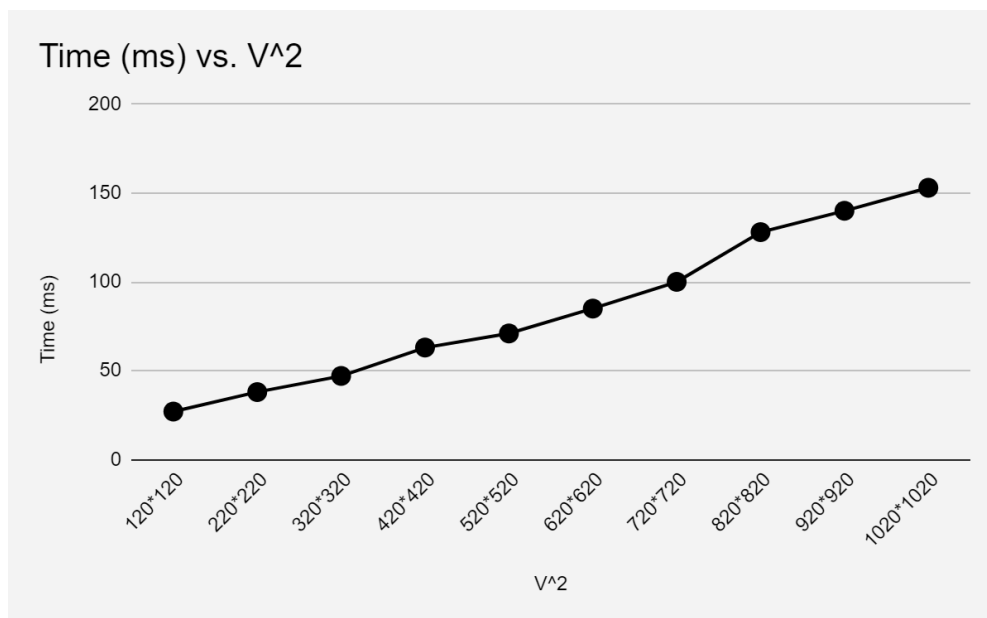
Grupo: TP062

Aluno: Francisco Ferro Pereira (107502)

-Complexidade global da solução: A complexidade global do algoritmo é $O(n^2)$, isto porque a leitura de entrada é $O(n^2)$, o algoritmo Kosaraju-Sharir $O(n^2)$, o DFS para encontrar o caminho mais longo $O(n^2)$ e a apresentação de dados no stdout $O(1)$. A complexidade seria $O(V+E)$ se utilizássemos uma lista de adjacências para representar o grafo invés de uma matriz de adjacências.

Avaliação Experimental dos Resultados

Para a avaliação experimental foram gerados 10 testes com tamanho de input incrementável. Alguns testes contêm grafos com SCCs e outros não. Foram temporizadas as execuções e traçado o seguinte gráfico.



Como podemos observar, ao pôr o eixo Ox a variar com a quantidade prevista na análise téorica, verificamos que existe uma relação linear com os tempos no eixo Oy, o que corrobora a nossa análise téorica de $O(V^2)$.