Relatório 1º projecto ASA 2020/2021

Grupo: al030

Aluno(s): Mara Alves (95625) e Margarida Rodrigues (95627)

Descrição do Problema e da Solução

Este projeto consiste em desenvolver um programa para conseguir determinar qual o número mínimo (**k**) de dominós que o Professor João Caracol deve derrubar de forma a garantir que todos os dominós das suas sequências caiam e, adicionalmente, o número de peças pertencentes à maior sequência de dominós a cair (**l**). As peças e as suas dependências de queda podem ser representadas por um DAG.

Para resolver este problema, assumimos que cada peça é identificada por um número de 0 a n-1 (sendo **n** o número de peças/vértices) para facilitar as indexações nos vetores, pois utilizámos um vetor para guardar o número de setas a apontar para esse vértice e outro para manter os vértices a que podemos chegar através de cada um. Começámos por ordenar topologicamente as peças de dominó utilizando o algoritmo de Kahn, com recurso a uma fila de espera em que guardamos sequencialmente as peças que ficam com o número de setas a zero. As primeiras peças a serem adicionadas a esta fila são as peças que devem ser derrubadas, e por isso o número de peças nesta fila corresponde ao número mínimo de dominós necessários derrubar. De seguida, percorremos as peças por ordem topológica e fomos relaxando as distâncias (guardadas num outro vetor) da peça às restantes a que consegue chegar, guardando o valor máximo da distância possível para cada dominó. O número de peças da sequência mais longa obtém-se somando um à maior distância, uma vez que a distância diz respeito às arestas, e o número de vértices é igual às arestas mais um.

Fontes:

- Kahn's algorithm for Topological Sorting
- Longest path in a directed acyclic graph (DAG)

Análise Teórica

- Leitura dos dados de input: primeiro de uma só linha com os valores de n e m; de seguida um ciclo linearmente dependente de **m** (o número de arestas), em que faz simultaneamente o processamento das arestas para os dois vetores de *incoming* e *outgoing edges.* Logo, Θ(1+m).
- Identificação dos vértices sem *incoming edges*: percorrer todo o vetor das *incoming edges* para obter os índices que têm o valor a zero. Logo, O(n).
- Apresentação do valor de k. Logo, O(1).
- Aplicação do algoritmo de Khan para obter a ordem topológica: ciclo em que se percorrem todos os vértices adjacentes a cada um dos vértices. Logo, O(n+m).
- Relaxamento das *outgoing edges* de todos os vértices, por ordem topológica, guardando no vetor das distâncias e numa variável os valores de distância mais altos. Logo, O(n+m).
- Apresentação do valor de l. Logo, O(1).

Complexidade global da solução: O(n+m).

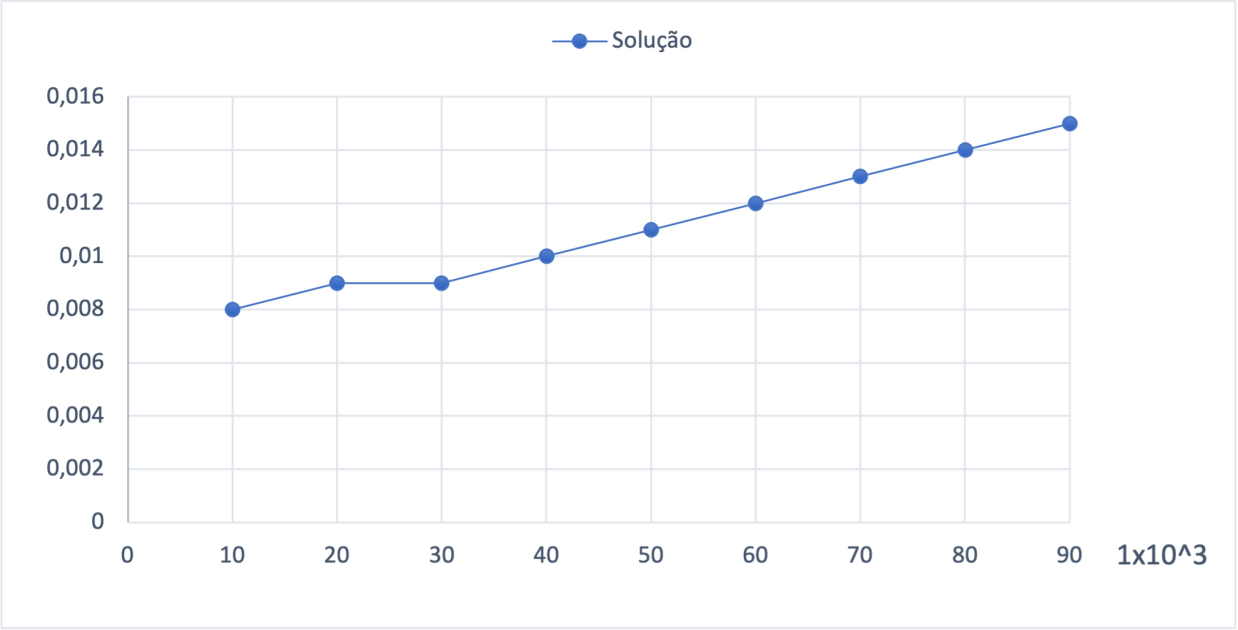
Relatório 1º projecto ASA 2020/2021

Grupo: al030

Aluno(s): Mara Alves (95625) e Margarida Rodrigues (95627)

Avaliação Experimental dos Resultados

Corremos o programa com diferentes ficheiros de input, gerados pelo algoritmo previamente fornecido, para testar a funcionalidade deste mesmo e, utilizando o comando time, obtivemos os diferentes tempos de cada teste.



O gráfico gerado é linearmente crescente, ou seja, à medida que aumentamos o tamanho do grafo, o tempo de execução incrementa também e, por isso, está de acordo com a análise teórica prevista.