Relatório 2º projeto ASA 2025/2026

Grupo: AL069
Alunos: Afonso Sítima(114018) e Rafael Dias(115156)

Descrição do Problema e da Solução

As estradas equivalem às arestas e os cruzamentos aos vértices de um grafo direcionado acíclico (DAG), o que significa que existe pelo menos um vértice sem arestas direcionadas para ele e outro sem arestas que saem dele e que é possível organizá-lo por ordem topológica.
O facto de ser um DAG permite-nos aplicar o algoritmo de Kahn que organiza o grafo por ordem topológica. Este algoritmo baseia-se na existência de vértices com grau de entrada nulo. O algoritmo remove iterativamente esses vértices e as respectivas arestas, construindo uma ordem topológica válida para todos os vértices do grafo.
Para verificar os caminhos, passamos por todos os vértices pela sua ordem topológica, e pelos seus vértices adjacentes, com isto conseguimos saber o número de caminhos para cada um dos vértices conectados ao atual vértice da iteração, fazendo isto para todos os vértices do grafo obtemos o número total de caminhos de qualquer vértice A para qualquer vértice B do grafo.

Análise teórica

- Leitura de dados de entrada e construção do grafo (simultâneas): primeiro é lido o de número de vértices que o grafo tem (n), o número de camiões(m), a gama de camiões a ser usada (m1|m2) e o número de arestas do grafo (k) (O(1) já que o número de dados a ler nesta fase é constante). De seguida é alocado o espaço para a lista de adjacências que guarda todos os vértices para onde se pode ir a partir de certo vértice (O(n+1)) e para a matriz que vai guardar o número de caminhos para cada vértice ($O(n+1)^2$). Por último são lidas k linhas que representam as arestas do grafo e que são guardadas diretamente na lista de adjacências (O(k)), assim $O(n,k) = O(1+n+1+n^2+2n+1+k)$ ou seja aproximadamente $\mathbf{O(n^2+k)}$.

- Aplicação do algoritmo para cálculo do valor pedido: aplicação do algoritmo de Kahn O(n+k). De seguida, num loop de n iterações, são processados todas as arestas ordenadas pela ordem topológica da origem da sua origem (O(k)) e é colocado na matriz o valor da soma do número de caminhos para o atual vértice (O(1)) e os caminhos adicionados para o vértice atual são distribuídos pelos respectivos camiões (O(n)), assim **O(n(k+n))**.

```
para i ← 1 até n fazer
    caminhos[i][i] ← 1
    para cada y em order fazer
        para cada ant em adj[y] fazer
            caminhos[i][ant] ← caminhos[i][ant] + caminhos[i][y]
            se caminhos[i][ant] > m então
                caminhos[i][ant] ← caminhos[i][ant] − m
            fim se
        fim para
    fim para
    para b ← 1 até n fazer
        se i = b ou caminhos[i][b] = 0 então
            continuar
        fim se
        num_cam ← 1 + (caminhos[i][b] mod m)
        se m1 ≤ num_cam ≤ m2 então
            adicionar o par (i, b) a pares[num_cam]
        fim se
    fim para
fim para
```

Relatório 2º projeto ASA 2025/2026

Grupo: AL069

Alunos: Afonso Sítima(114018) e Rafael Dias(115156)

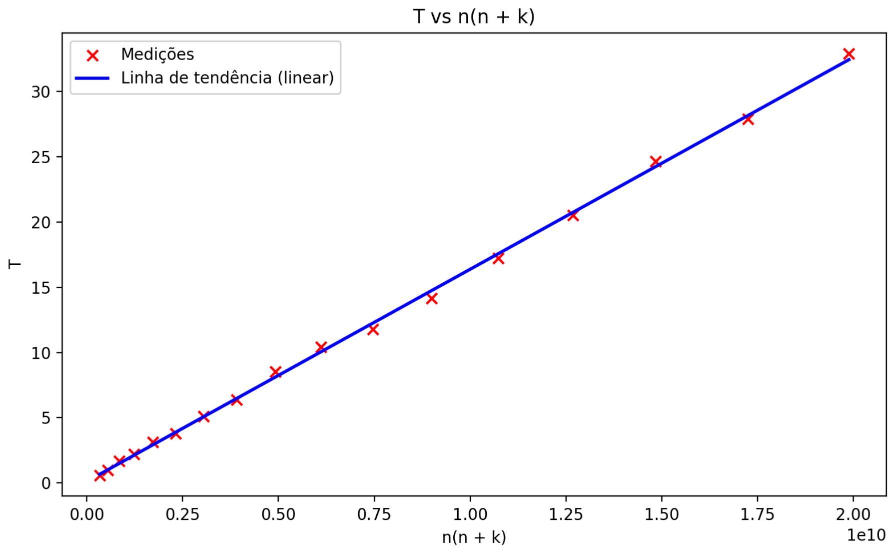
- Escrita de dados: o pior caso é O(n(n - 1)/2 aproximadamente O(n^2)) em que teria de escrever todos os caminhos, para qualquer outro caso dependerá dos valores de m2 e m1 e do número total de caminhos que cada camião de número i (m1<=i<=m2).

- Complexidade global da solução: aproximadamente igual à sua porção com maior complexidade, que neste caso é **O(n(k+n))**.

Avaliação Experimental dos Resultados

Para realizar a avaliação experimental, foram efetuados 17 testes com n a variar entre 1100 e 4300 com incrementos de 200. Já k é calculado com base numa probabilidade de 50% de uma aresta existir face ao número máximo de arestas possíveis num DAG. Logo k vai ser aproximadamente 0.5 * (n(n - 1)) / 2.

Eixo X: n(n + k) (complexidade) | Eixo Y: Tempo de execução (em segundos).



Como podemos observar, o gráfico de tempo em função de n(n + k) apresenta uma tendência linear, confirmando a complexidade O(n(n + k)) que previmos.