

# Relatório 2º projeto ASA 2025/2026

Grupo: AL096

Aluno(s): Bruna Almeida(114305) e João Varanda(114010)

## Descrição da Solução

De modo a auxiliar a empresa Entregas Caracol Lda., foi desenvolvido um algoritmo aplicado sobre um grafo que permite determinar as rotas dos camiões selecionados. Começa por se construir um grafo onde cada nó representa um cruzamento e cada aresta um caminho unidirecional entre dois centros de distribuição. Este grafo é armazenado numa lista de adjacências.

Relativamente ao algoritmo, a solução inicialmente calcula a ordenação topológica do grafo utilizando o algoritmo de Kahn, sendo que cada nó é processado apenas depois dos que o antecedem. Posteriormente, para cada nó é aplicada programação dinâmica sobre a ordenação topológica para determinar o número de caminhos possíveis entre um dado cruzamento origem e todos os restantes cruzamentos destino. Este número é obtido propagando o número de caminhos ao longo dos caminhos que começam nesse cruzamento. Para cada rota, é determinado o camião a usar através da fórmula:

$$\text{NúmeroCamiaoParaCaminho}(A,B) = 1 + \#\text{caminhos}(A,B)\%M,$$

sendo  $\#\text{caminhos}(A,B)$  o número de caminhos possíveis entre os centros de distribuição A e B e M o número total de camiões.

Para concluir, as rotas são agrupadas por camião atribuído, sendo apenas considerados os camiões dentro da gama fornecida.

## Análise Teórica

- Considerando:
  - N: número de cruzamentos (nós);
  - K: número de caminhos (arestas);
  - M: número total de camiões;
- Leitura de dados:  $O(N + K)$ . A leitura inclui a inicialização da lista de adjacências e de todos os cruzamentos.

---

**Cálculo das rotas**

```
1: function SOLVE( $N, M, m_1, m_2, paths$ )
2:    $topOrder \leftarrow \text{TOPOLOGICALORDER}(N, paths)$ 
3:   Initialize  $truckRoutes[0 \dots (m_2 - m_1)]$  as empty lists
4:   for  $i = 1$  to  $N$  do
5:     Initialize  $dp[1 \dots N] \leftarrow 0$ 
6:      $dp[i] \leftarrow 1$ 
7:     for all  $u \in topOrder$  do
8:       if  $dp[u] > 0$  then
9:         for all  $v \in paths[u]$  do
10:           $dp[v] \leftarrow (dp[v] + dp[u]) \bmod M$ 
11:        end for
12:      end if
13:    end for
14:    for  $t = 1$  to  $N$  do
15:      if  $t \neq i$  and  $dp[t] > 0$  then
16:         $truck \leftarrow \text{CALCULATETRUCKTOPATH}(M, dp[t])$ 
17:        if  $m_1 \leq truck \leq m_2$  then
18:          Append  $(i, t)$  to  $truckRoutes[truck - m_1]$ 
19:        end if
20:      end if
21:    end for
22:  end for
23:  return  $truckRoutes$ 
24: end function
```

---

# Relatório 2º projecto ASA 2025/2026

Grupo: AL096

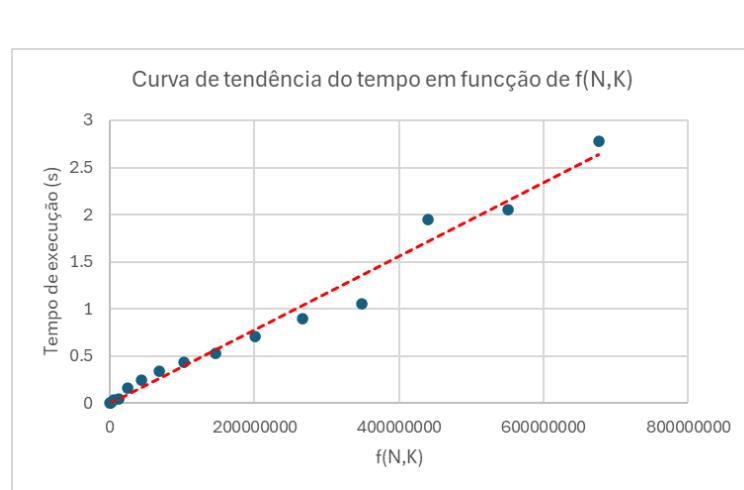
Aluno(s): Bruna Almeida(114305) e João Varanda(114010)

- Construção dos grafos:  $O(N + K)$ . O grafo é construído durante a leitura.
- Cálculo do número de caminhos e atribuição de camiões:  $O(N \cdot (N+K))$ . Utilizando o algoritmo de Khan, são percorridos todos os cruzamentos para calcular o grau de cada um e criar um vetor ordenado topologicamente ( $O(N + K)$ ). Este vetor é atravessado N vezes, 1 por cada cruzamento.

Assim, a complexidade global do algoritmo é  **$O(N(N + K))$** , pois sobrepõe-se a todas as outras complexidades inferiores. Todas as outras operações, contribuem com uma complexidade com uma velocidade de crescimento inferior.

## Avaliação Experimental dos Resultados

Para realizar esta avaliação foram realizados 15 testes, variando N de 100 a 1500 com incrementos de 100 e com um valor de  $m_1$ ,  $m_2$  e  $M$  constantes iguais a 10, 40 e 50, respetivamente. A probabilidade de existência de caminhos foi mantida a 40%.



$f(N,K)$	Tempo (s)
207800	0.004
1623400	0.007
5506500	0.041
12967600	0.045
25290500	0.159
43621200	0.243
68973800	0.342
102710400	0.433
146317500	0.527
201201000	0.709
266490400	0.901
348193200	1.051
440144900	1.949
550435200	2.053
676164000	2.775

O gráfico obtido apresenta um comportamento linear quando o tempo é representado em função de N e de K, isto é,  $f(N,K) = N(N + K)$ . Assim é possível confirmar a hipótese de que o algoritmo cresce de acordo com a análise teórica.

DONE