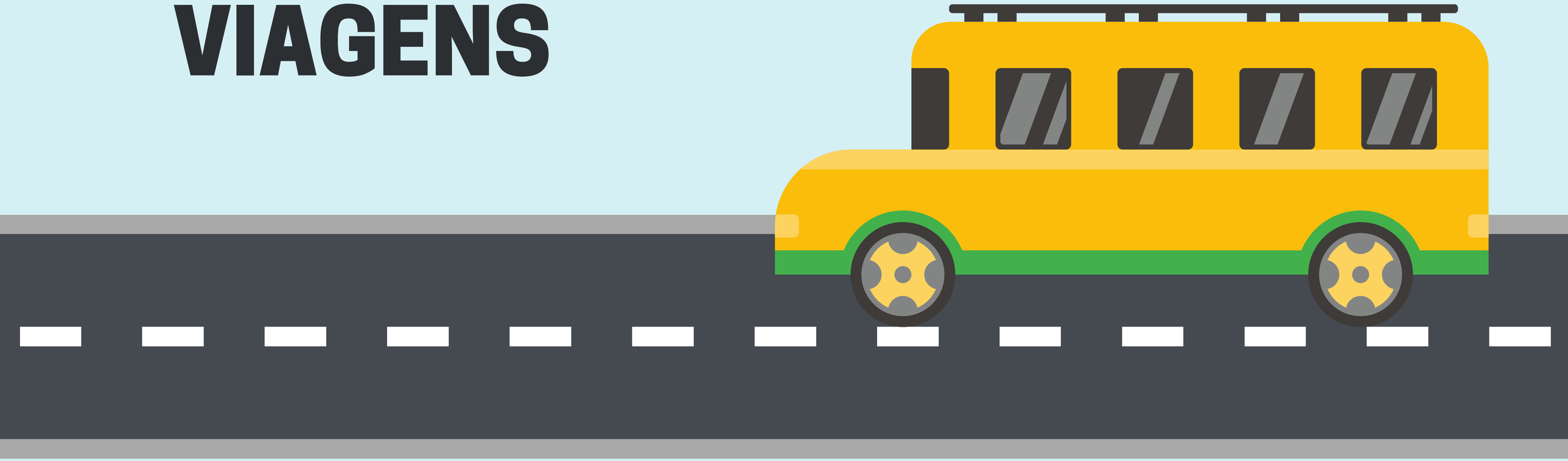
AGÊNCIA DE VIAGENS

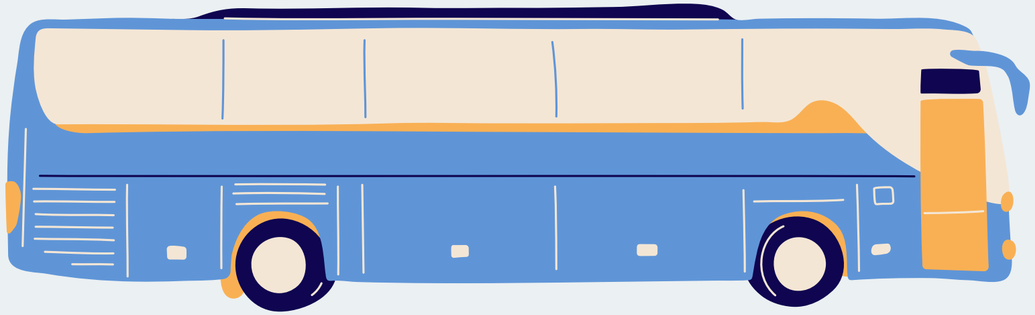


Diogo Fonte - up202004175
João Tedesco - up202000416
Sofia Moura - up201907201

DESCRIÇÃO DO PROBLEMA

O objetivo principal deste trabalho é implementar um sistema digital para uma agência de viagens, capaz de apoiar a gestão de pedidos para transporte de grupos de pessoas de um local de origem para um local de destino, ambos dados inseridos pelo utilizador.
Para que tal seja possível, vão ser implementadas e exploradas diferentes funcionalidades no sistema com o objetivo de maximizar o número de passageiros, minimizar o número de transbordos ou o tempo de espera, tendo em conta se o grupo de passageiros pode ou não separar-se.

CENÁRIO 1 (GRUPOS QUE NÃO SE SEPARAM)



FORMALIZAÇÃO DOS PROBLEMAS

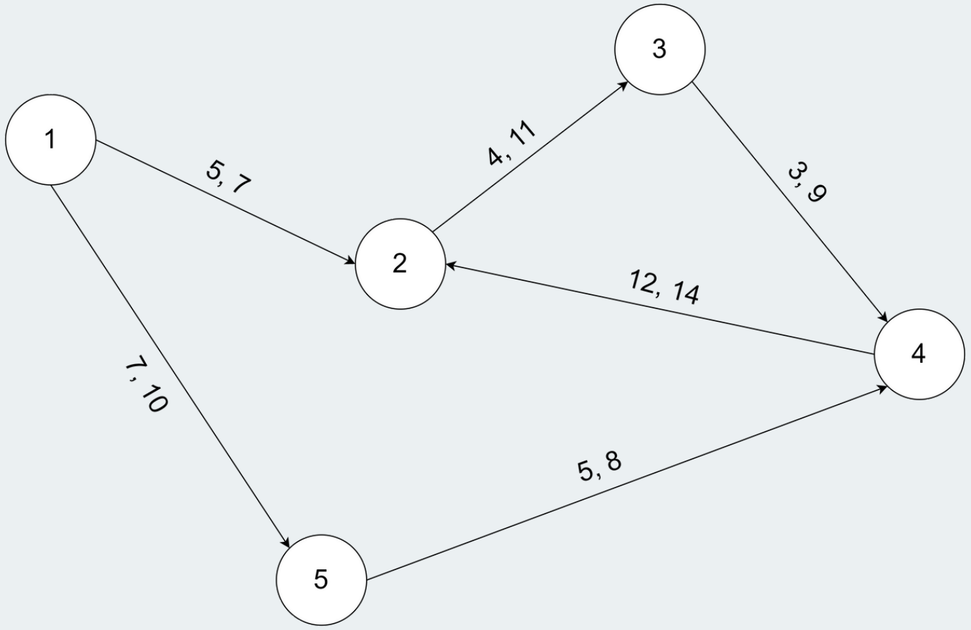
Dados:

- o - nó de partida (origem)
- d - nó de chegada (destino)
- n_1 ... n_m - nós
- a_1 ... a_p - arestas
- cap_1 ... cap_p - capacidade das arestas

Variáveis de decisão:

- x_1 ... x_n - nós da resposta

SELEÇÃO E IMPLEMENTAÇÃO DE ESTRUTURAS DE DADOS



Para cada ficheiro, decidimos criar um grafo dirigido. Os nós representam as paragens (cada número corresponde ao inteiro apresentado nos ficheiros para representar internamente cada paragem da classe **Grafo**) e as arestas representam o caminho que é percorrido entre os nós que ligam (guardam 2 números que correspondem à capacidade do veículo responsável por esse trajeto e a duração da viagem, respetivamente).

IMPLEMENTAÇÃO DE ALGORITMOS

Para resolver as alíneas 1 e 2 do cenário 1, entendemos que o melhor seria utilizar o algoritmo Dijkstra. Para a alínea 1, foram necessárias algumas adaptações do algoritmo, como a utilização das capacidades das arestas para calcular a capacidade do percurso, isto é, o mínimo das capacidades dos ramos que o constituem.

Para a alínea 2, demos novamente uso ao algoritmo anterior para mostrar a maximização da dimensão do grupo, no entanto, para minimizar o número de transbordos utilizamos o algoritmo clássico de Dijkstra.

ANÁLISE DE COMPLEXIDADES

1.1
Complexidade temporal: O(A * log N)

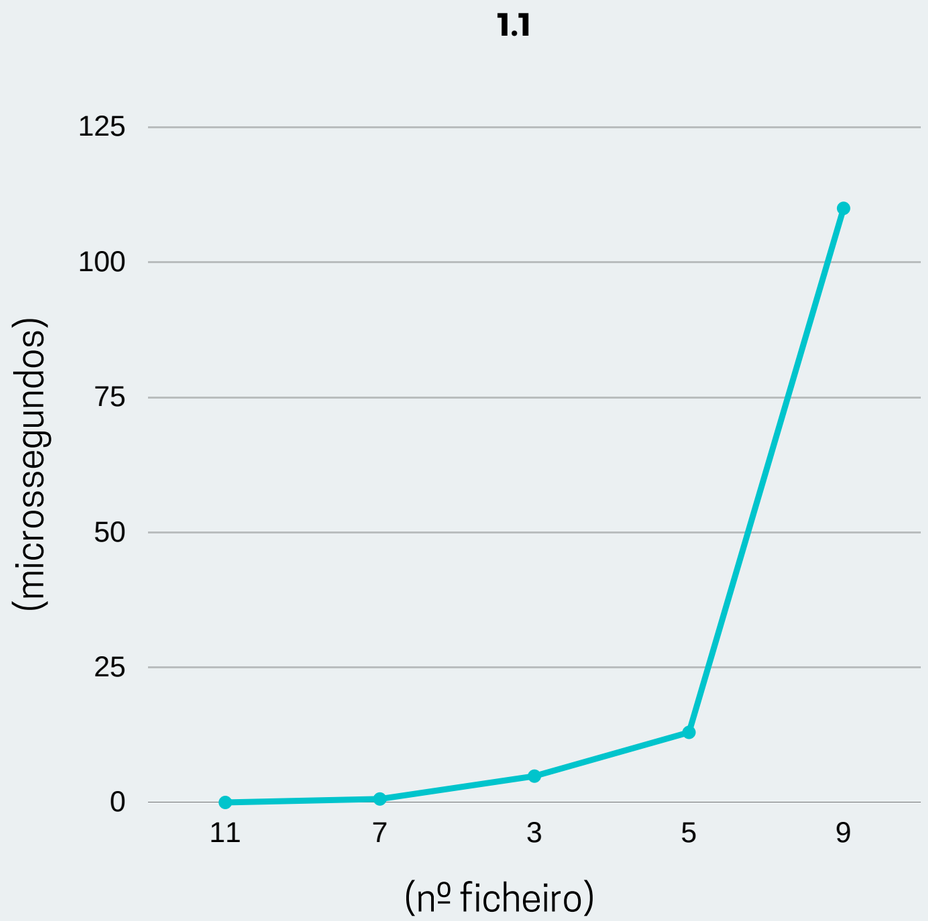
Complexidade espacial:
- priority_queue<pair<int,int>>: O(N)

1.2
Complexidade temporal: O(|A| log |N|)

Complexidade espacial:
- MinHeap<int, int>: O(N)

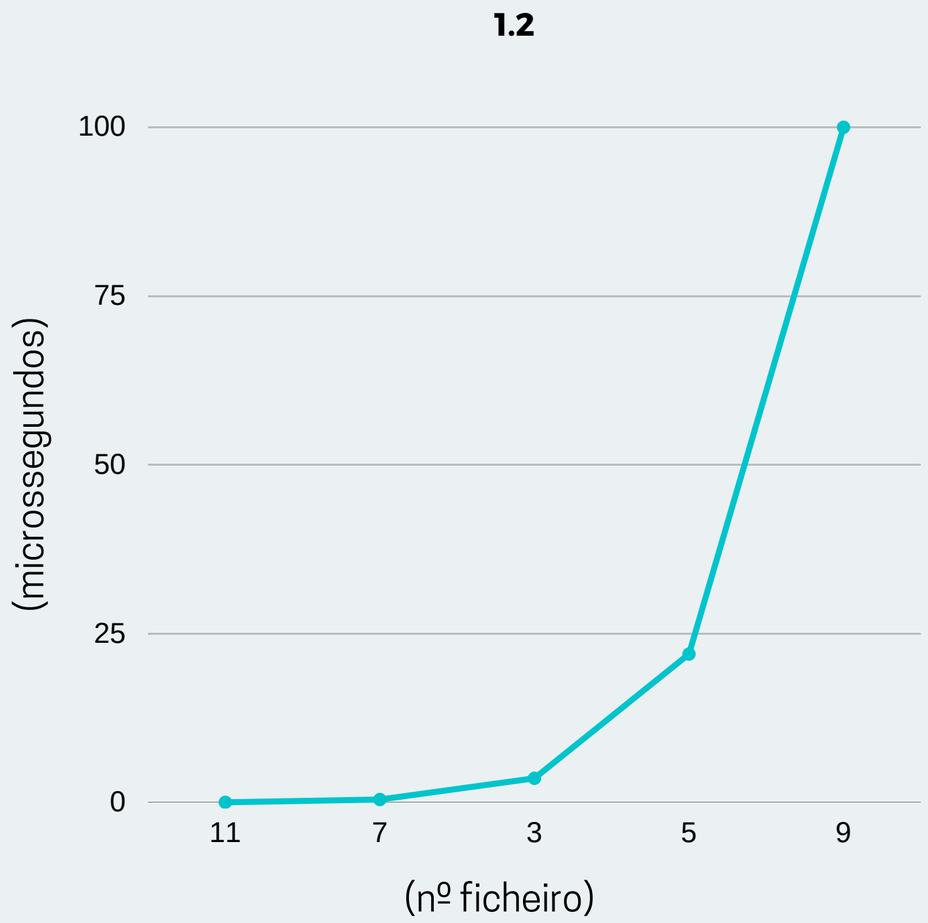
obs: N corresponde aos nós do grafo e A, às suas arestas

AVALIAÇÃO EMPÍRICA



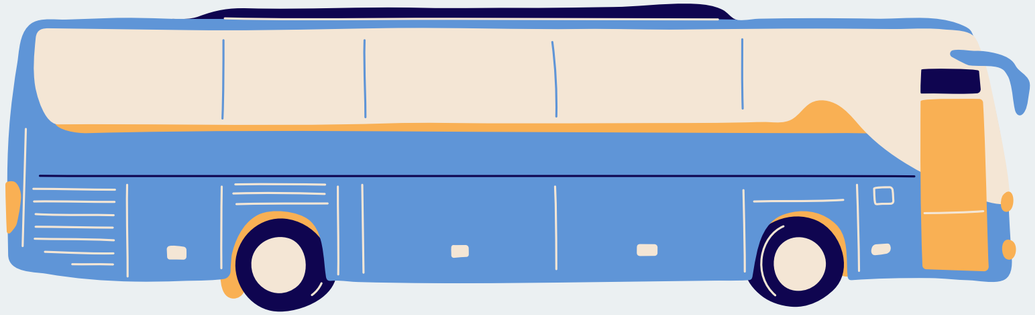
Conclusão: Podemos concluir que a complexidade temporal do algoritmo implementado para solução desta alínea é linear. O gráfico tem uma forma diferente do esperado porque as dimensões dos ficheiros de input testados possuem variações de dimensão bastante acentuadas. A variação de dimensão entre os ficheiros 11 e 3 são menores do que entre os ficheiros 3 e 9 (4 - 90 - 300 - 1000 - 5000 nós).

AVALIAÇÃO EMPÍRICA



Conclusão: Podemos concluir que a complexidade temporal do algoritmo implementado para solução desta alínea é linear. O gráfico tem uma forma diferente do esperado porque as dimensões dos ficheiros de input testados possuem variações de dimensão bastante acentuadas. A variação de dimensão entre os ficheiros 11 e 3 são menores do que entre os ficheiros 3 e 9 (4 - 90 - 300 - 1000 - 5000 nós).

CENÁRIO 2 (GRUPOS QUE SE PODEM SEPARAR)



FORMALIZAÇÃO DOS PROBLEMAS

Dados:

- o - nó de partida (origem)
- d - nó de chegada (destino)
- n_1 ... n_m - nós
- a_1 ... a_p - arestas
- cap_1 ... cap_p - capacidade das arestas
- dur_1 ... dur_p - duração das arestas

Variáveis de decisão:

- x_1 ... x_n - nós da resposta

SELEÇÃO E IMPLEMENTAÇÃO DE ESTRUTURAS DE DADOS

	1	2	3	4
1	0	5	5	2
2	0	0	6	3
3	0	0	0	7
4	2	0	0	0

Além da classe **Grafo** usada no cenário anterior, decidimos representar a informação de forma diferente para certos algoritmos do cenário 2, uma vez que a classe anterior dificultava o acesso às capacidades de um nó para o outro. Representamos o grafo num vetor bidimensional de inteiros (correspondente à matriz de adjacências), guardando-o como atributo da classe **Grafo2**, onde o elemento vetor[i][j] refere-se à capacidade do percurso de do nó i para o nó j. Para os nós que não têm ligação, o valor do elemento do vetor correspondente é 0.

IMPLEMENTAÇÃO DE ALGORITMOS

2.1. / 2.2. / 2.3. Algoritmo de Edmonds-Knap para maximizar a dimensão do grupo com a sua separação, análogo ao algoritmo de Ford Fulkerson, mas que procura o menor caminho possível, utilizando o algoritmo de pesquisa em largura, BFS.

2.4. Método do caminho crítico para determinar a duração mínima do percurso de um nó para outro e o tempo de início mais próximo para cada aresta

2.5. Método do caminho crítico para obter o tempo de conclusão mais afastado para o percurso com fim no nó desejado, por *backward analysis*.

ANÁLISE DE COMPLEXIDADES

2.1. / 2.2. / 2.3.
Complexidade temporal:

- ciclo while(G > 0): O(G)
  - ciclo for(): O(C)
  - ciclo for(): O(C)
    - reverse: O(N(do path))
    - ciclo for(): O(N(do path))

Complexidade espacial:

- List<int> paths: O(C)
- List<int> caps: O(C)
- Array booleano visited: O(N)

2.4.
Complexidade temporal: O(|N| * |A|^2)

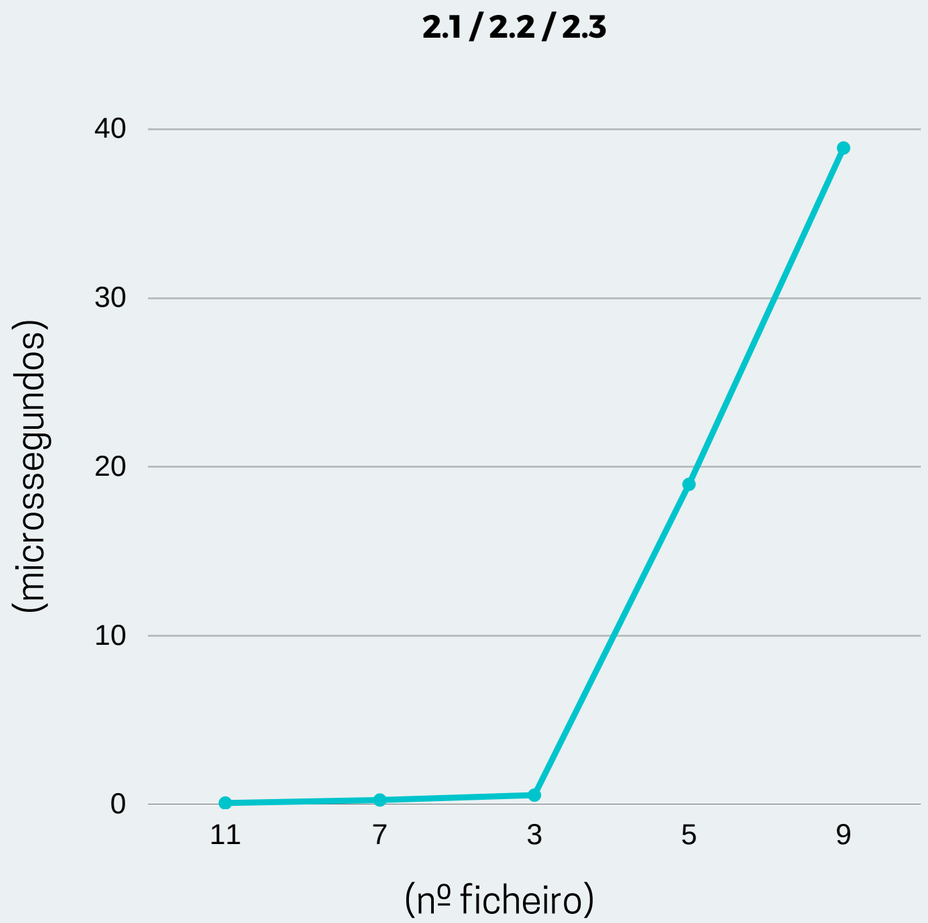
Complexidade espacial: O(|N|^2)

2.5.
Complexidade temporal: O(|N| * |A|^2)

Complexidade espacial: O(|N|^2)

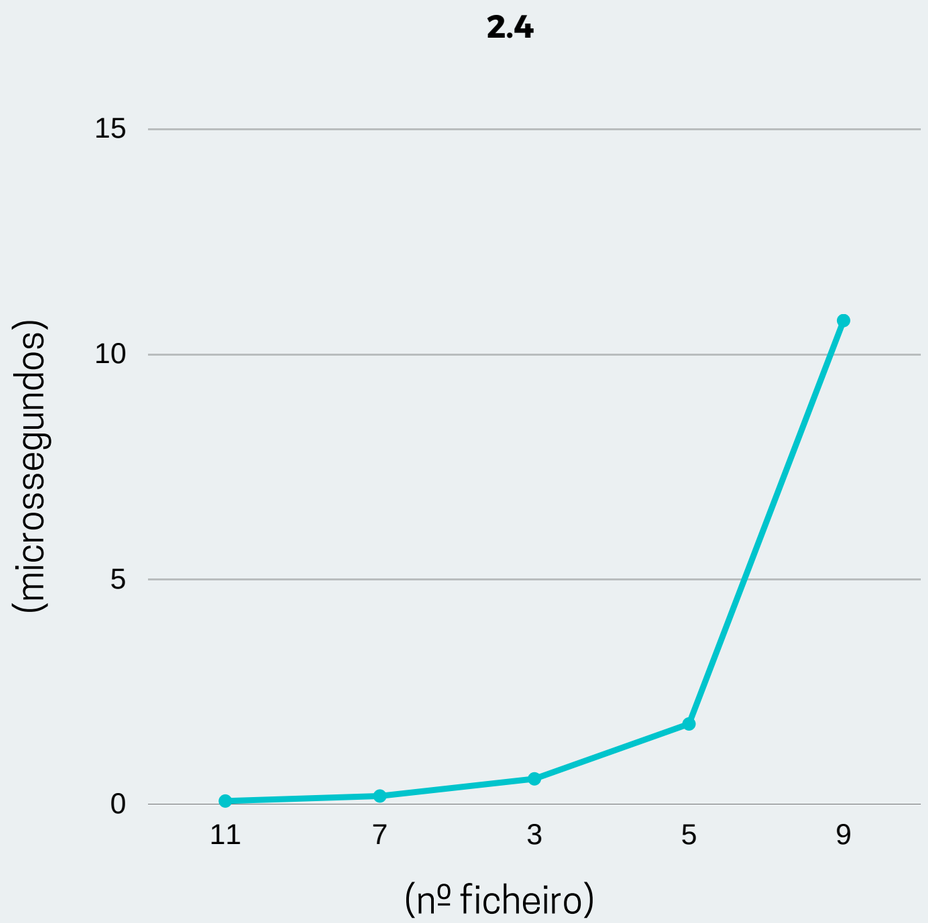
obs: N corresponde aos nós do grafo, A às arestas, G à dimensão do grupo e C ao número de caminhos encontrados.

AVALIAÇÃO EMPÍRICA



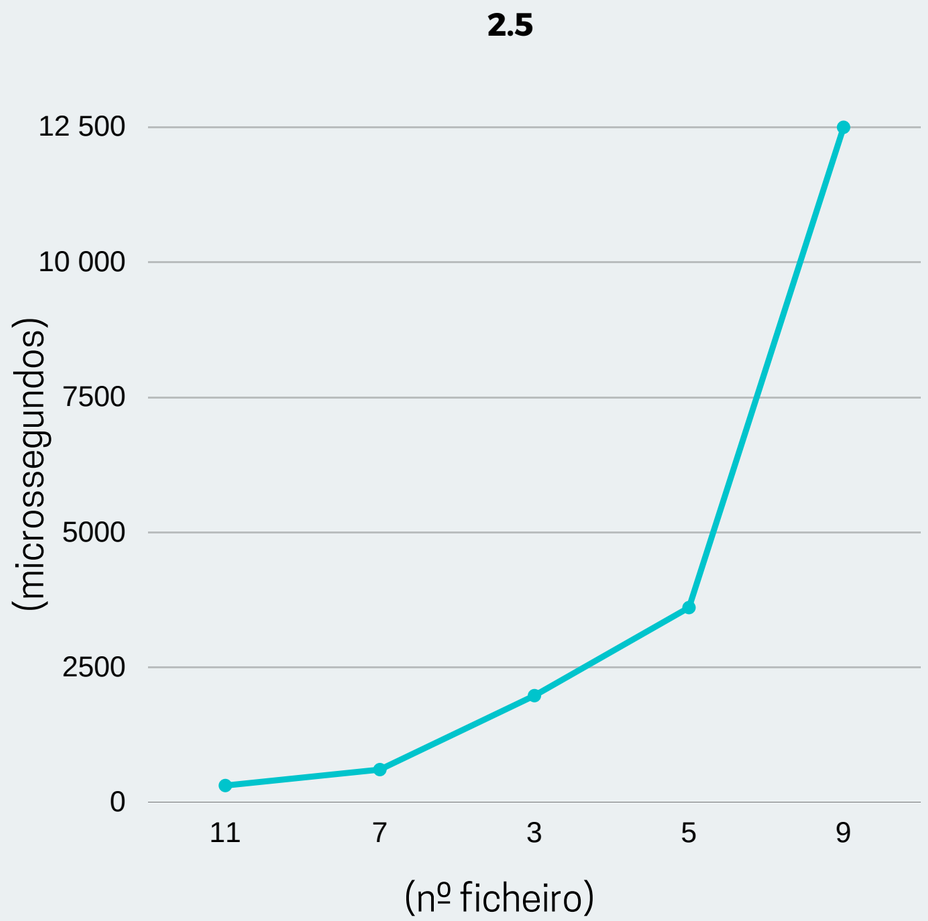
Conclusão: Podemos concluir que a complexidade temporal do algoritmo implementado para solução destas 3 alíneas é linear. O gráfico tem uma forma diferente do esperado porque as dimensões dos ficheiros de input testados possuem variações de dimensão bastante acentuadas. A variação de dimensão entre os ficheiros 11 e 3 são menores do que entre os ficheiros 3 e 9 (4 - 90 - 300 - 1000 - 5000 nós).

AVALIAÇÃO EMPÍRICA



Conclusão: Podemos concluir que a complexidade temporal do algoritmo implementado para solução desta alínea é exponencial.

AVALIAÇÃO EMPÍRICA



Conclusão: Podemos concluir que a complexidade temporal do algoritmo implementado para solução desta alínea é exponencial.

ESFORÇO E PRINCIPAIS DIFICULDADES

DIFICULDADES

- Planeamento dos algoritmos utilizados para cada opção de percurso
- Pensamento conceptual das soluções das alíneas apresentadas

ESFORÇO DOS ELEMENTOS DO GRUPO

- As tarefas foram divididas por cada elemento do grupo de forma equilibrada e todos se esforçaram igualmente