

# Relatório 1º projeto ASA 2021/2022

**Grupo:** al086

**Aluno(s):** Allan Fernandes (97281)

---

## Descrição do Problema e da Solução

O problema apresentado subdivide-se em dois subproblemas, problema 1 e problema 2, o objeto do problema 1 é o cálculo do tamanho da maior subsequência estritamente crescente ( $t$ ) e igualmente o número de subsequências de tamanho máximo ( $c$ ), ambos dados uma sequência de inteiros. O problema 2 consiste no cálculo do tamanho da maior subsequência comum estritamente crescente entre duas quaisquer sequências de inteiros. Relativamente a implementação, a solução foi desenvolvida na linguagem c++.

Para o problema 1 optou-se pela utilização de programação dinâmica, segundo o *bottom-up approach*<sup>1</sup>, para tal foi necessário utilizar duas estruturas, representadas no código como *vector*, sendo que, o primeiro serve para guardar o tamanho máximo da subsequência que acaba num índice  $i$ -th e o segundo tem como intuito salvaguardar o número de subsequências cujo tamanho encontra-se associado ao primeiro *vector* e que acaba num índice  $i$ -th, o tamanho dessas estruturas corresponde ao tamanho duma sequência fornecida, estas foram inicializadas a 1, já que, cada inteiro numa posição  $j$ -th quando visto individualmente satisfaz a condição de estritamente crescente. Percorre-se a sequência comparando o inteiro na posição  $i + 1$  com o inteiro na posição  $i$ , caso se satisfaça a condição de “estritamente crescente”, num primeiro caso, atualiza-se um conjunto de variáveis relativas ao tamanho da subsequência atual e da anterior, num segundo caso, testa-se a igualdade entre as variáveis supracitadas, caso iguais, incrementa-se o número de subsequências associadas ao índice atual, num terceiro e último caso, é verificado se o tamanho da subsequência atual é maior do que a anterior, se satisfeito obteve-se agora, uma nova subsequência estritamente crescente de maior dimensão, caso ainda não se tenha atualizado o tamanho da variável referente a subsequência de maior dimensão possível, esta é atualizada, seguidamente, é atualizado os valores relativos ao tamanho máximo da subsequência que acaba num índice  $i$ -th e o número de subsequências associadas ao mesmo índice  $i$ -th. Por último percorre-se o *vector* de tamanho máximo, quando encontra um índice, tal que o tamanho máximo ( $t$ ) é igual ao valor associado ao *vector* de tamanho máximo, incrementa-se a variável referente ao número de subsequências de tamanho máximo ( $c$ ).

Para o problema 2 utilizou-se igualmente uma solução de programação dinâmica. A ideia desta solução passa por guardar num *vector* de dimensão igual a segunda sequência pós um pré-processamento (no qual só foi guardado elementos que já existem na primeira sequência), este foi inicializado a zero, uma vez que, num pior caso não há quaisquer elementos comuns entre as duas sequências. Este *vector* guarda o tamanho da maior subsequência comum estritamente crescente que acaba num índice  $i$ -th, sendo este último limitado pelo tamanho da segunda sequência. Percorre-se todos os elementos da primeira sequência, e para cada elemento desta, percorre-se os elementos da segunda sequência, no caso eventual de encontrar elementos de igual valor em ambas, atualiza-se o valor associado a respetiva entrada no *vector*.

---

<sup>1</sup> começa-se por resolver pequenos subproblemas e as suas soluções são posteriormente utilizadas para chegar a solução de subproblemas maiores.

# Relatório 1º projeto ASA 2021/2022

Grupo: al086

Aluno(s): Allan Fernandes (97281)

---

## Análise Teórica

**Problema 1** - Seja  $N$ , o tamanho da sequência fornecida.

- Leitura dos dados de entrada: simples leitura do input para um *vector*.  $O(1)$
- Aplicação do algoritmo para a obtenção do tamanho da maior subsequência estritamente crescente ( $t$ ), ciclo for (2 no total) itera sobre o tamanho da sequência.  $O(N^2)$
- Apresentação dos dados.  $O(1)$

Complexidade global da solução:  $O(N^2)$

**Problema 2** - Considere  $N$ ,  $M$  e  $L$ , onde  $N$  é o tamanho da primeira sequência,  $M$  o tamanho da segunda,  $L$  tamanho da segunda sequência após um pré-processamento (num pior caso é igual a  $N$ ).

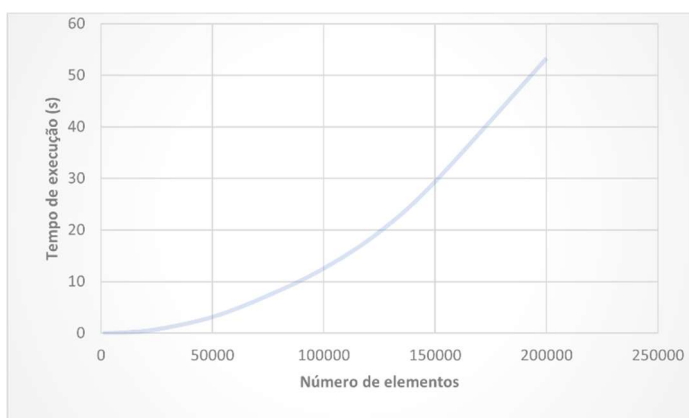
- Leitura dos dados de entrada e processamento do segundo vetor (inclusão apenas dos comuns).  $O(1)$
- Aplicação do algoritmo para a obtenção do tamanho da maior subsequência comum estritamente crescente ( $t$ ).  $O(N^2)$  ou  $O(NL)$
- Apresentação dos dados.  $O(1)$

Complexidade global da solução:  $O(NL)$

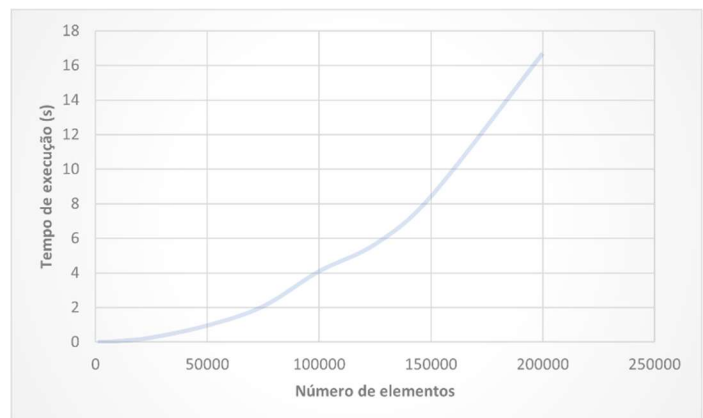
## Avaliação Experimental dos Resultados

Com o intuito de validar os resultados teóricos acima, foram geradas 10 instâncias de teste de tamanho incremental para cada um dos problemas, problema 1 e 2.

Problema 1



Problema 2



Para o problema 1 foram geradas instâncias de tamanho correspondente a (1000,5000,10 000,25 000,50 000,75 000,100 000,125 000,150 000,200 000). Quanto ao problema 2 a instâncias geradas correspondem a (1000 – 1200, 5000 – 5200, 10 000 – 10 200, 25 000 – 25 200, 50 000 – 50 200, 75 000 – 75 200, 100 000 – 100 200, 125 000 – 125 200, 150 000 – 150 200, 200 000 – 200 200).

É possível ver o caráter curvilíneo de ambos os gráficos, portanto, a análise teórica é satisfeita. Segue-se que a complexidade de ambas as soluções é quadrática.