Relatório 1º projecto ASA 2021/2022

Grupo: al116

Aluno(s): Margarida Bezerra (99270) e Maria Sofia Pinho (99272)

Descrição do Problema e da Solução

Problema 1: Dada uma sequência de inteiros pretende-se calcular o tamanho da sua maior subsequência estritamente crescente e o número de subsequências estritamente crescentes de tamanho máximo.

Solução 1: O algoritmo utilizado recebe um vetor com os inteiros da sequência, **sequence**, e o seu comprimento, **len**. São criados mais dois vetores com comprimento **len**: um para armazenar em cada posição **i** o número das maiores subsequências estritamente crescentes que começam no valor **sequence[i]**, chamado **contador_seq_i**, com todos os elementos iniciados a 0 exceto o último com valor 1 (para sinalizar que há pelo menos uma); e outro para armazenar em cada posição **i** o tamanho da maior subsequência estritamente crescente que começa no valor **sequence[i]**, chamado **tamanho_i**, com todos os elementos iniciados a 1. Para preencher estes vetores percorre-se a sequência da direita para a esquerda e, em cada iteração, vai-se comparar o valor atual com todos os valores à sua direita, havendo assim um loop exterior, loop 1, e um loop interior, loop 2. Sempre que um valor à **direita** (**j**) for maior do que um valor à **esquerda** (**i**), se o tamanho das maiores subsequências crescentes que começam no valor à direita (**tamanho_i[j]**) + 1 for maior que o número das que começam no valor à esquerda (**tamanho_i[i]**), então, há um novo tamanho de maior sequência crescente, e o número destas é igual ao número de maior sequencias crescentes que começam no valor da direita (porque à maior sequência crescente da direita, estamos a adicionar um valor à esquerda, menor que todos os outros), logo, altera-se o valor de **tamanho_i[i]** para **tamanho_i[j] + 1** e o valor de **contador_seq_i[i]** para **contador_seq_i[j]**. Por outro lado, se **tamanho_i[j] + 1** for igual a **tamanho_i[i]**, adiciona-se o número de maiores subsequências crescentes do da direita ao número de maiores subsequências crescentes ao da esquerda (**contador_seq_i[i] += contador_seq_i[j]**). Por fim, o tamanho da maior subsequência estritamente crescente é o maior valor de **tamanho_i** e o número de subsequências estritamente crescentes deste tamanho é a soma de todos os **contador_seq_i[i]** onde, **tamanho_i[i]** é o valor máximo.

(**Nota:** no código, devido a um lapso na atualização dos nomes das variáveis, foram trocados os nomes dos vetores **tamanho_i** e **contador_seq_i**)

Problema 2: Dadas duas sequências de inteiros pretende-se calcular o tamanho da maior subsequência comum estritamente crescente.

Solução 2: o algoritmo utilizado recebe dois vetores, cada um com os inteiros das sequências 1 e 2, **seq1** e **seq2**, e os seus respetivos comprimentos, **len1** e **len2**. Começa-se por criar o vetor **len_subseq** com comprimento **len2** e todos os elementos a 0, onde, em cada posição **i** do vetor, vai ser guardado o maior comprimento das subsequências estritamente crescentes comuns entre **seq1** e os valores da 2ª até **seq2[i]**. Para preencher **len_subseq** percorre-se a **seq2** (**j** iterações) para cada elemento de **seq1** (**i** iterações), havendo uma variável auxiliar **current** que é iniciada a 0 para cada iteração da **seq1**, que terá o valor do comprimento da maior subsequência comum até **seq2[j]** no momento. Quando os valores a ser analizados, **seq1[i]** e **seq2[j]**, são iguais e **current + 1** é maior do que **len_subseq[j]** (de momento o maior compriment armazenado até **seq2[j]**) o valor é atualizado para **current + 1**. Quando **seq1[i]** é maior do que **seq2[j]** e **len_subseq[j]** é maior que **current**, ou seja, quando o valor a ser considerado de **seq1** é maior do que um valor da **seq2** que já foi identificado anteriormente como igual a um valor anterior na mesma sequência, é atualizado o current, para, caso no futuro **seq1[i]** seja igual a um elemeno seguinte da **seq2**, o current ter o valor da correspondente subsequência estritamente crescente descoberta. Após isto, é percorrido o

Relatório 1º projecto ASA 2021/2022

Grupo: al116

Aluno(s): Margarida Bezerra (99270) e Maria Sofia Pinho (99272)

---

len_subseq para encontrar o seu maior valor, que corresponde ao tamanho da maior subsequência estritamente crescente comum às duas sequências.

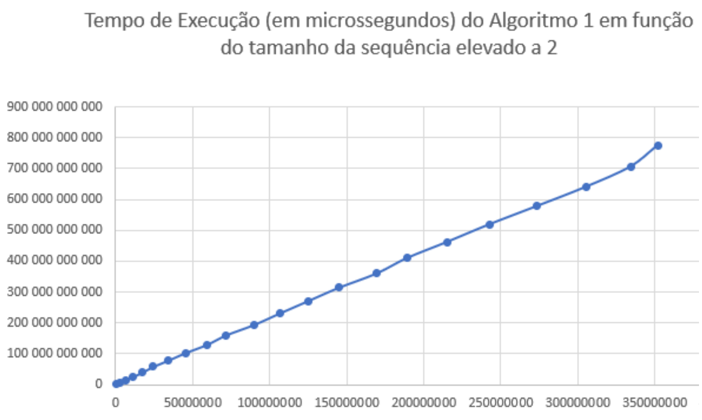
Análise Teórica

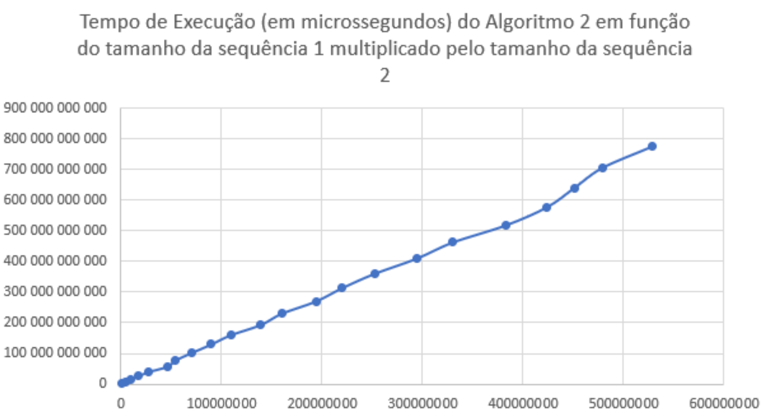
- Leitura de dados de entrada da solução 1: é utilizada a função getline() e um ciclo, ambos a depender dos caracteres do input (c) e de complexidade linear, assim tem complexidade, sendo n o número de valores na sequência, de 2.O(c) = 2.O(2n) = O(4n). Logo, **O(n)**;
- Leitura de dados de entrada da solução 2: o mesmo procedimento da solução 1, com o acrescimo da criação de um set não ordenado com os valores da primeira sequência, que tem custo em média de O(n), e um ciclo dependente dos número de inteiros da sequência 2, em que em cada iteração se procura o valor lido pelo input no set, o que tem complexidade O(1), tendo assim o ciclo O(n). Logo, **O(n)**;
- Aplicação do algoritmo 1: sendo n o número de elementos da sequência, são realizados dois ciclos, loop 1 e loop 2, estando o loop 2 dentro do loop 1, o 1 corre n vezes, enquanto o 2 corre n – i vezes, sendo i a iteração atual do loop 1, assim, a sua complexidade é O(n(n+1)/2), de seguida ocorre mais um ciclo com n iterações, ou seja, O(n). Logo, **O(n²)**;
- Aplicação do algoritmo 2: sendo n e m o número de elementos das sequências 1 e 2 respetivamente, é realizado um ciclo de n iterações em que em cada iteração ocorre outro ciclo com m iterações, tendo assim O(n*m), de seguida há mais um ciclo com m iterações, O(m). Logo, **O(n²)**;
- Apresentação os dados: é feita com com o cout para ambos os problemas. Logo, **O(1)**;

Complexidade global da solução 1: O(n²)

Complexidade global da solução 2: O(n²)

Avaliação Experimental dos Resultados





Os gráficos gerados estão concordantes com a análise teórica prevista.