Relatório 1º projecto ASA 2023/2024

Grupo: TP062
Aluno: Francisco Ferro Pereira (107502)

Descrição do Problema e da Solução

A solução proposta envolve determinar o valor máximo extraído de sub-retângulos de dimensões 1x1 até XxY, onde X é a largura e Y é a altura da chapa de mármore. Esta é uma abordagem do tipo bottom-up, pois resolvem-se primeiro problemas menores de forma ótima e mais tarde o problema original. Para guardar os preços das peças é utilizado um array bidimensional. Para guardar os resultados ótimos dos sub-problemas, é utilizada uma matriz de inteiros. Utiliza-se a técnica de "memoization" para evitar a recomputação de soluções já conhecidas. Posteriormente, itera-se pela matriz. A cada iteração, popula-se a entrada da matriz com o maior lucro possível para um retângulo com as respectivas dimensões. Esse valor pode ser obtido através de uma peça com as mesmas dimensões que o retângulo a considerar ou por um corte que seja mais lucrativo. Um corte é dividir um retângulo em dois menores. Calcular o seu valor é somar os valores ótimos dos dois sub-retângulos. A abordagem bottom-up com memoization facilita o acesso a esses valores. Por fim, repete-se este algoritmo para todas as dimensões até atingir a entrada XxY da matriz, onde se encontrará a solução ótima para o problema original.

Análise Teórica

- Leitura de input:
- O(1) para a leitura das dimensões da chapa e para o número de peças.
- A leitura das dimensões e preços das peças depende do número de peças (N). Esta tarefa é O(N), ou seja, linear.
- Inicializar o array bidimensional para guardar os preços custa O(X*Y). Esta tarefa cresce quadraticamente com as dimensões da chapa de mármore, sendo por isso o fator dominante na leitura de input. A inserção de preços no array é O(1).

Em suma, a complexidade temporal da leitura de input é **O(X*Y)**.

- Função recursiva:
- Inicializar a matriz de memoization tem um custo de O(X*Y).
- Os nested loops iteram pela matriz de memoization inteira, tendo um custo de O(X*Y).
- A cada iteração calculamos o número de cortes. Estes cálculos custam O(1).
- A cada iteração acedemos ao array para procurar o preço de uma potencial peça correspondente. O acesso a elementos deste array é O(1).
- A cada iteração a exploração de cortes custa O(X/2 + Y/2) onde X/2 é o número de cortes verticais e Y/2 o de cortes horizontais.

Em suma, a complexidade temporal da função recursiva é **O(X*Y*(X/2+Y/2)).**

- Apresentação de dados: Escrever o resultado no stdout é O(1).

Relatório 1º projecto ASA 2023/2024

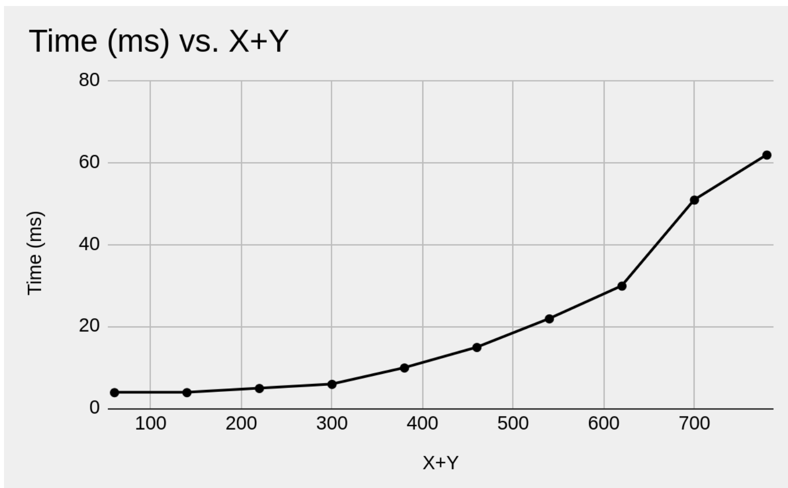
Grupo: TP062

Aluno: Francisco Ferro Pereira (107502)

-Complexidade global da solução: A complexidade do algoritmo é $O(n^3)$, isto porque a leitura de entrada é O(X*Y), a função recursiva O(X*Y*(X/2+Y/2)) e a apresentação de dados O(1).

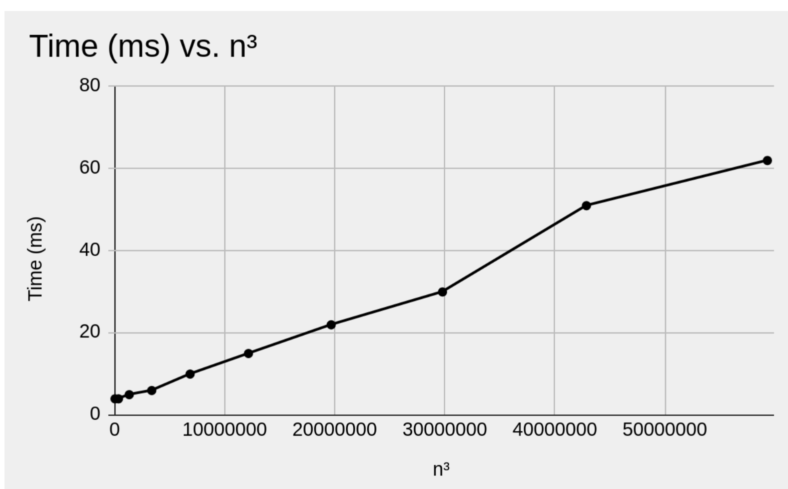
Avaliação Experimental dos Resultados

Para a avaliação experimental foram gerados 10 testes com tamanho de input incrementável. Foram temporizadas as execuções e traçados os seguintes gráficos.



Como podemos observar, o tempo de execução não é linear com as dimensões da chapa (X+Y).

Assim, vamos por o eixo dos XX a variar com a quantidade prevista pela proposta feita na análise teórica ($O(n^3)$), obtendo o seguinte gráfico:



Ao mudarmos o eixo dos XX para n^3, verificamos que existe uma relação linear com os tempos no eixo dos YY, confirmando que a nossa implementação está de acordo com a análise teórica de $O(n^3)$.