Relatório 1º projeto ASA 2024/2025

Grupo: AL070

Aluno(s): Ana Santos (109260) e Francisco Mendonça (109264)

Descrição do Problema e da Solução

O problema consiste, dada uma tabela que define um operador binário sobre inteiros e uma sequência de aplicações sucessivas do operador associada a um resultado, em descobrir a ordem pela qual efetuar a parentização da sequência de modo a obter o resultado.
A nossa solução para o problema baseia-se na construção de uma tabela bidimensional com dimensão $(m \times m)$, preenchida através da recursão. Cada célula da tabela é composta por um vetor de pares em que o primeiro elemento é o valor gerado por cada parentização e o segundo é um vetor que permite reconstruí-la.

Análise Teórica

- **Pseudo-código:**

```
 john(left,right)
|  min = left
|  max = right
|  let values be a vector of pairs
|  while (right>min)
|  |  x = john(min, right-1)
|  |  y = john(right,max)
|  |  decrement right
|  |  for i=0 to size of x
|  |  |  for j = 0 to size of y
|  |  |  |  values = (table[x[i]][y[j]], [right, first element of x[i], first elemento of y[j]])
|  calculator[left][max] = values
|  return values
```

Exemplo do preenchimento de uma célula:

	1	2
1	2	1
2	1	2

Sequência: 1 2 1 2 1

CALCULATOR(2,4) = [[1, [2, 1, 2]], [1, [1, 2, 1]]]

<u>Observações:</u> A string é reconstruída a partir de outra função recursiva que divide a sequência nas subsequências que resultam no valor pretendido. Quando os resultados são encontrados coloca cada parte entre parênteses.

- Leitura dos dados de entrada: simples leitura do input, com um ciclo a depender linearmente do tamanho da sequência de inteiros (m) e outro ciclo a depender quadraticamente do tamanho do conjunto de inteiros (n^2). Logo, $O(n^2 + m)$.
- Aplicação do algoritmo indicado para cálculo do valor que tem o número máximo de m^2 chamadas recursivas onde, o cálculo de cada célula tem o custo do ciclo que percorre todas as possíveis divisões (m) onde cada divisão tem associado três ciclos de tamanho máximo n (n^3). Logo, $(m^2) \times (n^3 \times m) = O(n^3 \times m^3)$.
- Aplicação do algoritmo para reconstrução da string com o valor de $m + (m-1) \Rightarrow 2 \times m \Rightarrow m$, chamadas recursivas onde o custo de cada chamada envolve um ciclo de tamanho máximo n. Logo, $O(n \times m)$.
- Apresentação dos dados: $O(1)$.

Complexidade global da solução: $O(n^3 \times m^3)$

Relatório 1º projecto ASA 2024/2025

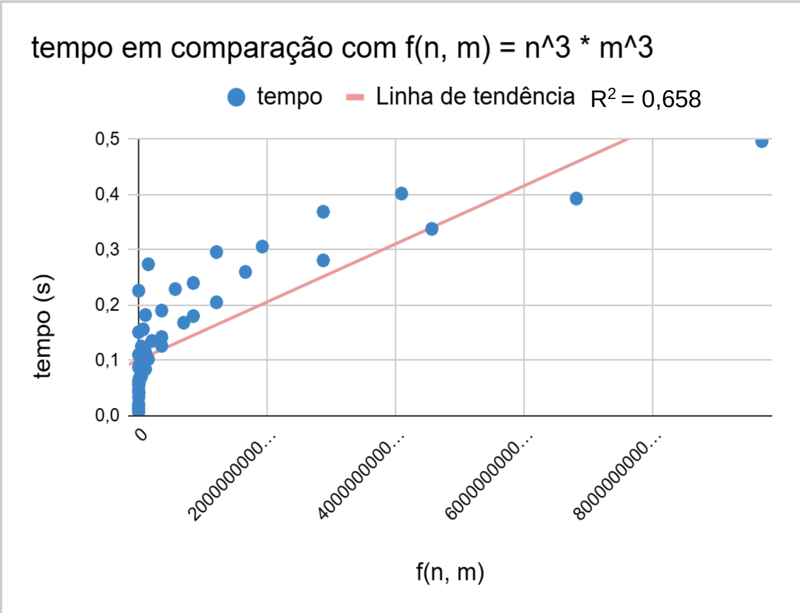
Grupo: AL070

Aluno(s): Ana Cláudia Santos (109260) e Francisco Mendonça (109264)

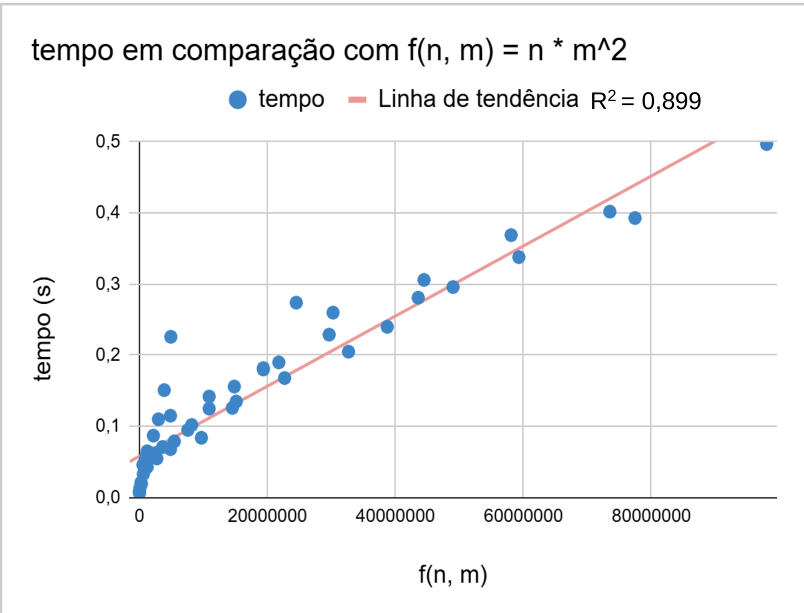
Avaliação Experimental dos Resultados

Para a parte experimental utilizamos o gerador de instâncias fornecido por forma a testar a análise de complexidade teórica. Geramos 50 instâncias de tamanho incremental $f(n, m)$ com diferentes combinações de n (5 -100) e m (10-1000) e medimos o tempo de execução.

Para a análise teórica $O(n^3 \times m^3) \Rightarrow f(n, m) = n^3 \times m^3$, o seguinte gráfico representa o tempo (eixo dos YYs) em função de $f(n, m)$ (eixo dos XXs).



Ao analisar o gráfico, é possível verificar que a análise teórica da complexidade do nosso algoritmo não está correta pois não é linear em relação ao tempo (R^2 é muito menor que 1), isto pode ter acontecido devido a diversos fatores, tais como o número de chamadas recursivas ser muitas vezes menor que o máximo, para além de não serem realizadas as 3 iterações de tamanho n na maioria dos casos, devido às diversas condições de paragem implementadas.



Assim, o $f(n, m)$ que se aproxima mais aos testes realizados e torna a relação observada próxima a linear é $n \times m^2$, ou seja, $O(n \times m^2)$. Como é possível observar no seguinte gráfico com R^2 próximo de 1.