Relatório 1º projecto ASA 2024/2025

Grupo: TP032

Aluno(s): Diogo Fernandes (110306) e Michael Andrew (63484)

Descrição do Problema e da Solução

O problema da parêntesação mais à esquerda para uma dada expressão com um operador definido por uma matriz de valores foi resolvido utilizando uma abordagem de programação dinâmica *bottom-up*, cujo preenchimento pode ser simplificado em:

$$\boldsymbol{dp}(i,j) = \boldsymbol{MatrizDeValores}[\boldsymbol{i-1}][\boldsymbol{i-1}], \quad \boldsymbol{i == j}$$

$$\boldsymbol{dp}(i,j) = \cup(\boldsymbol{MatrizDeValores}[\boldsymbol{a-1}][\boldsymbol{b-1}]), \ \boldsymbol{j > i}$$

tal que a união decresce de x = j até i + 1, a ∈ dp(i, x-1) e b ∈ dp(x,j).

No caso da solução submetida, cada célula da matriz tem estruturas "dp_item" que contém não só o valor da expressão, representado pela fórmula acima, como também os valores da direita e da esquerda da expressão, e o índice de parêntesação. Estes elementos são necessários para, posteriormente, construir recursivamente a solução de uma forma eficiente (espaço e tempo).

O preenchimento da matriz garante que os resultados são atingidos por ordem da parêntesação mais à esquerda. Como tal, foram desenvolvidas otimizações: armazenamento apenas da primeira ocorrência de cada resultado possível para cada célula da matriz e interrupção de cálculos de sub-problemas assim que todas as respostas possíveis para este são alcançadas. Um vetor de *booleans* garante que estas verificações sejam feitas em tempo constante.

Análise Teórica

```
expressãoParentesada Solution(){
        Input(n, m, valueMatrix, Sequence, Answer);
        // MartrixSize = n, SequenceSize = m
        DPMatrixConstruction();
        expressãoParentesada = RecursiveBuildBrackets();
        return expressãoParentesada;
}
```

- **Leitura dos dados de entrada**: leitura de *n* e *m*, de uma matriz de valores *n* por *n*, de uma sequência de inteiros de tamanho m e respetivo resultado: O(1) + O(n^2) + O(m) + O(1). Logo, **O(n^2 + m).**
- **Construção da Matriz de programação dinâmica**: **O($m^3 \times n^2$)** Diagonal principal: O(m). Restantes Diagonais: $\sum_{k=1}^{m-1}\sum_{i=0}^{m-k} \boldsymbol{kn^2} \cong \boldsymbol{m^3n^2}$. Expressão deduzida do código submetido. Utiliza nomenclaturas de variáveis usadas no mesmo, excetuando *m* e *n*. Intuição: preenchimento de uma matriz de lado *m* através da combinação de *m* células e da combinação de *n* valores de uma célula com n valores de outra.
- **Construção recursiva da parêntesação: O($n \times m$)**. Para cada uma das *m* divisões feitas sobre a sequência, são percorridos *n* elementos.
- **Apresentação dos dados:** Inclui a função recursiva e um *loop* O(n). A chamada à função recursiva só é feita uma vez no máximo. Logo **O(n) + O($n \times m$)** ≅ **O($n \times m$)**.

Complexidade Global da Solução: **O(m^3n^2).**

Relatório 1º projecto ASA 2024/2025

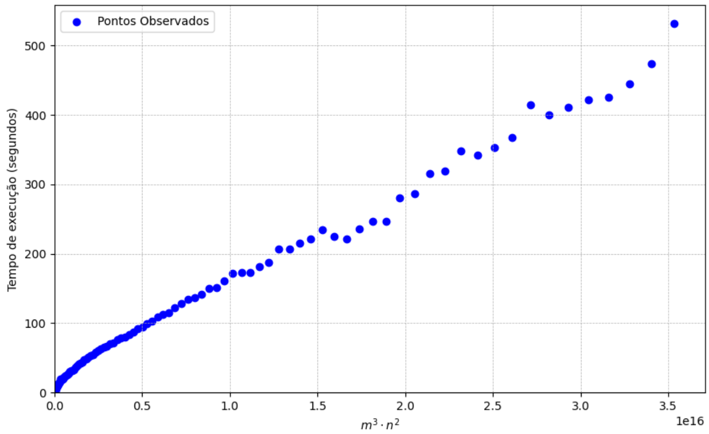
Grupo: TP032

Aluno(s): Diogo Fernandes (110306) e Michael Andrew (63484)

Avaliação Experimental dos Resultados

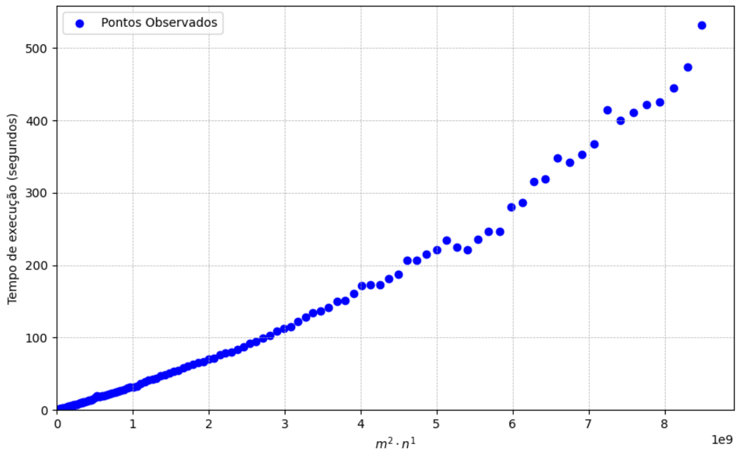
Foram simuladas boas condições de temperatura, bateria e houve desativação de processos desnecessários. Foi integrado ao projeto, através da biblioteca *chrono*, um temporizador para medir o tempo de execução após a geração incremental dos inputs (de 15 em 15). Observaram-se 136 pontos no total. Os tempos registados foram posteriormente utilizados para construir os gráficos.

O gráfico seguinte relaciona a complexidade teórica, $O(m^3n^2)$, com os tempos de execução. A tabela apresenta apenas alguns pontos observados.



m	*n*	Tempo / s
15	15	0.000075
300	300	0.579939
600	600	6.253210
750	750	13.234625
900	900	23.100072
1200	1200	60.738326
1680	1680	206.698374
1860	1860	319.197725
1995	1995	425.144565
2040	2040	531.788450

Na fase teórica, foi calculado um limite assimptótico superior (O), pelo que o gráfico observado não pode ser equiparado diretamente a uma reta. Como no melhor caso possível, as células da matriz são preenchidas com *n* valores logo nas primeiras iterações do algoritmo, o que faz com que não sejam também necessárias *m* combinações de células para preencher uma só célula, estabelece-se o limite assimptótico inferior $\Omega(m^2n)$.



Raramente são atingidos o melhor e o pior caso na prática. Apesar de não ser definido o caso intermédio, conclui-se que a complexidade do algoritmo está entre as curvas dos dois gráficos apresentados.