

Relatório 1º projecto ASA 2024/2025

Grupo: AL030

Aluno(s): Francisco Pestana (ist1109625)

Descrição do Problema e da Solução

O problema consiste na criação de um algoritmo que determine como adicionar parênteses em uma sequência de inteiros (ex. 2 2 2 2 1 3) de forma a obter um resultado desejado, usando operações definidas por uma tabela de operadores binários. O algoritmo utiliza duas matrizes $s \times s$, sendo s a dimensão da sequência de elementos a colocar parêntesis:

Ambas as matrizes são de entradas $x(i,j)$ onde i e j são dois índices de uma sub-sequência (por exemplo, para $i = 1$ e $j = 3$, a sub-sequência seria 2 2 2).

1. **Matriz de Resultados:** Para cada subsequência delimitada por índices i, j regista os possíveis valores que podem ser obtidos.
2. **Matriz de Detalhes:** Armazena informações das combinações, como o valor gerado, os operandos e o índice de divisão

Para resolver o problema, primeiro é preenchida a diagonal principal da matriz com os casos de sub_sequências unitárias, em seguida, são explorados todas as combinações de tamanho dois (2) até m (tamanho da sequência). Isto garante que a matriz é preenchida pelas diagonais. Em seguida, explora subsequências crescentes até cobrir o intervalo completo, dividindo-as em duas partes com base em um índice de divisão. Para cada divisão são iteradas todas as combinações possíveis na parte esquerda e na parte direita. Cada combinação é registada como também os seus dados. Ao final, verifica se o resultado desejado está presente na subsequência completa. Caso esteja, reconstrói a expressão utilizando a matriz de detalhes para identificar as divisões e operandos, e uma função recursiva de reconstrução que se chama recursivamente a ela própria tanto do lado esquerdo como direito utilizando o índice de divisão. Se não for possível encontrar o resultado, a saída indica essa impossibilidade.

Análise Teórica

- **Leitura do Input:**
 - Ler n e s $O(1)$
 - Ler tabela $n \times n$ $O(n^2)$
 - Ler sequência $O(s)$
 - Ler o resultado a encontrar $O(1)$
 - Total:** $O(n^2 + s)$
- **Processamento de dados para o algoritmo principal:**
 - Criar matrizes $s \times s$ $O(s^2)$
 - Preencher diagonal principal das matrizes $O(s)$
 - Total:** $O(s^2)$
- **Algoritmo principal:**
 - for (size len = 2 to size len = s), $O(s)$
 - for (sub-sequence of size len) $O(s)$
 - for (each separation index i) $O(s)$
 - for (combination on left side) $O(n)$ (n resultados diferentes)
 - for (combination on right side) $O(n)$
 - calculate and register results

Total: $O(s^3 \times n^2)$

Relatório 1º projecto ASA 2024/2025

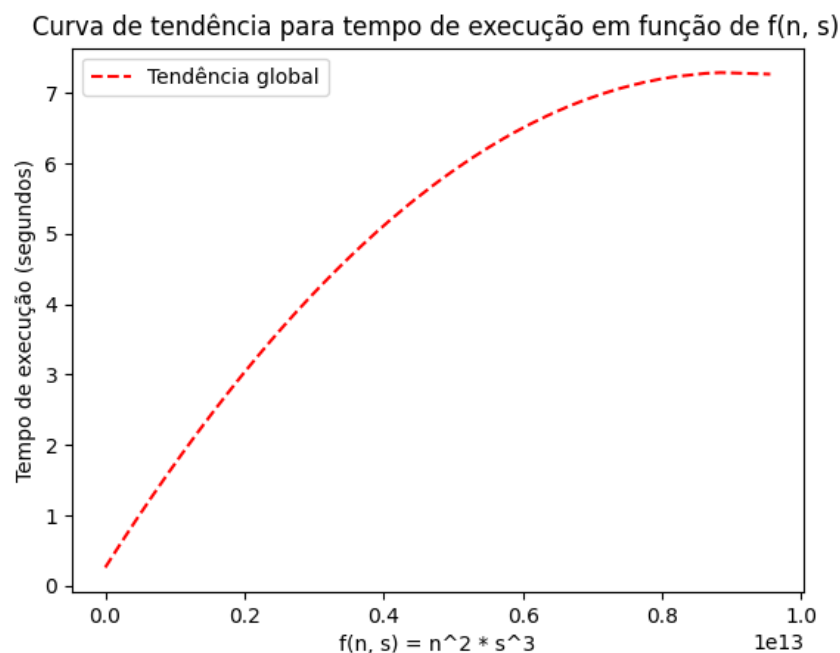
Grupo: AL030

Aluno(s): Francisco Pestana (ist1109625)

- **Reconstrução da expressão:**
for (results in details matrix) $O(n)$
 if (expected result)
 base case
 leftpart = recursive call to left side
 rightpart = recursive call to right side $O(s)$
 return (leftpart + rightpart)
Total: $O(s \times n)$
Complexidade global: $O(s^3 \times n^2)$

Avaliação Experimental dos Resultados

Foram cronometrados sucessivos inputs de dimensão crescente (matrizes de entrada de dimensão $5 \leq n \leq 100$, com acréscimos de 5; e sequencias de dimensão $10 \leq s \leq 1000$, com acréscimos de 25)



O gráfico apresenta certa lineariedade até um ponto, onde os problemas se tornam muito extensos, onde se vê uma “estabilização”. Esta “estabilização” resulta dos parametros de restrição de iterações/computações, os quais reduzem o tempo de execução, redução que é mais evidente quando os problemas são significativamente maiores (por exemplo, para uma tabela de entrada de dimensão 10, mas uma sequencia de dimensão 1000, cada entrada da tabela é operada só o número de vezes necessário até encontrar 10 valores diferentes)