

Relatório 1º projecto ASA 2022/2023

Grupo: TP021

Aluno(s): Henrique Costa (103663)

Descrição do Problema e da Solução

O problema era descobrir o número total de configurações de quadrados $N \times N$ tal que os quadrados não ultrapassem os limites estabelecidos pelo input.

Na minha solução consiste num loop com recursão em que primeiro são calculadas as configurações com 1×1 , após isso voltamos uma geração atrás e tentamos 2×2 e dps 3×3 até ao valor máximo Quadrado que foi calculado. Os valores são guardados num map para evitar a repetição de cálculos.

Análise Teórica

– Leitura dos dados de entrada: ler as dimensões linhas e colunas $O(1)$, com ciclo for para o número de linhas $O(\text{linhas})$. Logo $O(\text{linhas})$.

– Chamar função recursiva: $O(c^n)$

Função recursiva: $O(c^n)$

– Eliminar os primeiros zeros (primeiro while): $O(n)$

– Encontrar key no unordered_map: Average: constant, mas $O(n)$, $n = \text{map.size}()$

– Encontrar maior elemento no vetor: $O(n)$, $\text{vetor.size}()$

– Inserir par de vetor e número no unordered_map: Average: constant, mas $O(n)$, $n = \text{map.size}()$

– Obter primeiro valor de maxSquare: $O(1)$

– Ciclo while que diminui valor do quadrado máximo: $O(\text{quadradoMaximo})$

– Ciclo for que diminui o valor do quadrado máximo: $O(\text{quadradoMaximo})$

– Ciclo for que cria cópia de vetor e um vetor v3: $O(n)$, $n = \text{vetor.size}()$

– Ciclo for para os quadrados que cabem naquela posição: $O(\text{quadradoMaximo})$

– Ciclo for para retirar ao vetor o espaço ocupado pelo quadrado inserido: $O(\text{lado quadrado})$

– Chamar novamente Função recursiva: $O(c^n)$

– Ciclo for para reiniciar os vetores: $O(\text{lado quadrado})$

– Inserir par de vetor e número no unordered_map: Average: constant, mas $O(n)$, $n = \text{map.size}()$

– Limpar vetores: $O(n)$

– Complexidade global da solução: $O(c^n)$

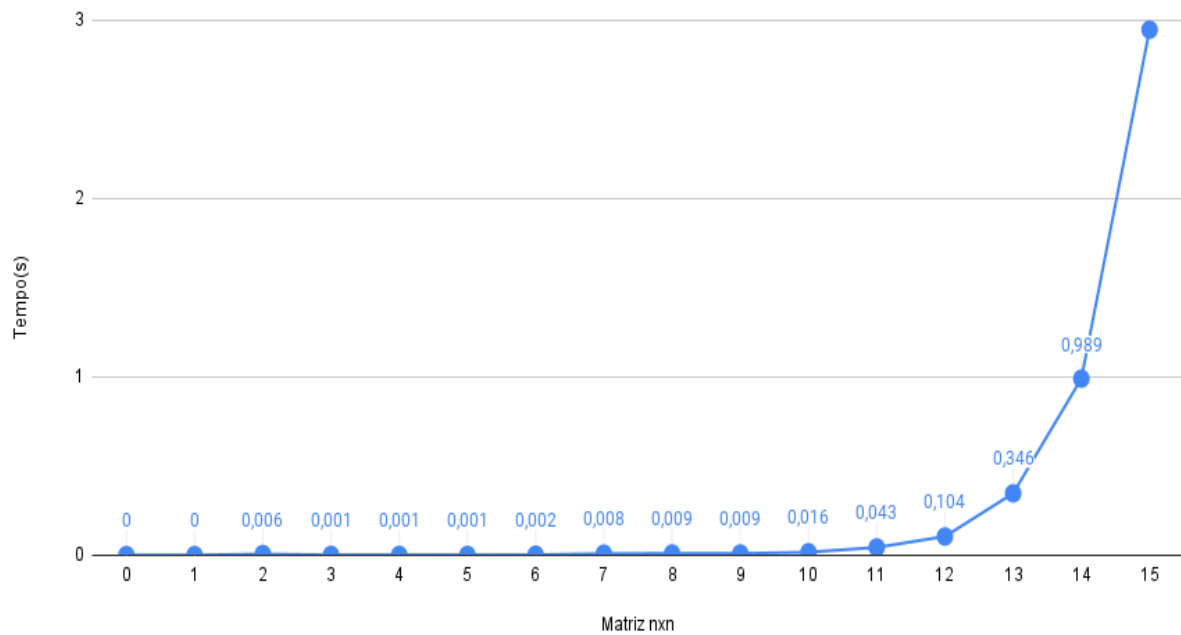
Relatório 1º projecto ASA 2022/2023

Grupo: TP021

Aluno(s): Henrique Costa (103663)

Avaliação Experimental dos Resultados

Tempo(ms) em comparação com Matriz nxn



Como se pode ver pelo gráfico foram geradas 15 experiências para as diferentes matrizes nxn.

Pelo gráfico a função que mais se aproxima é $0.2104e^{0.5354x}$ que nos diz que o nosso programa é $O(e^n)$ o que valida o que obtivemos na análise teórica de $O(c^n)$.