Relatório 1º projecto ASA 2022/2023

Grupo: AL057

Aluno(s): André Ferreira de Oliveira(103011) e Beatriz Mira Mendes(103120)

Descrição do Problema e da Solução

Para solucionar o problema proposto construímos uma matriz onde é desenhada a escada de acordo com o input inserido. De seguida, começando pela primeira linha no canto direito é visto o maior quadrado (NxN) que é possível desenhar, sendo retirado este quadrado e repetido o processo até que apenas sobre uma linha na matriz, indicando uma possibilidade. Chegando a este ponto, regressa à matriz inicial onde volta a ser testada a colocação de um quadrado, desta vez de dimensão (N-1xN-1) e repetindo o processo de eliminação de quadrados e linhas. Não sendo possível fazer mais iterações numa certa linha, esta é eliminada e reinicia-se o processo todo para a linha seguinte.

Desta forma, a contagem de possibilidades é feita de forma recursiva, em que cada vez que são feitas alterações à matriz é chamada a função de contagem com a nova matriz.

Análise Teórica

Na primeira etapa são recolhidos os inputs do utilizador e é criada a matriz a partir dos valores inseridos. Nesta fase, é realizado um primeiro loop Θ(n) para saber o número de linhas, onde para cada linha é realizado um loop com o input m do número da coluna do caminho, Θ(m) para preencher a matriz. Assim, a complexidade total da fase de leitura de inputs é de Θ(n*m). Pseudocódigo:

```
N = input;
For N do
    m = input;
    for m do
        preencheMatrix;
    end for;
end for;
```

A próxima etapa consiste na verificação de um int para confirmar se a matriz nao se encontra vazia, sendo a complexidade deste processo Θ(1) e no caso de se confirmar é impressa a resposta que também é de complexidade Θ(1).

Caso ultrapasse o passo anterior é chamada a função recursiva de contagem dos casos possíveis para solucionar o problema. Dentro da função recursiva são efetuados 2 loops, um de complexidade Θ(n) , que tem como objetivo calcular o tamanho máximo do lado de um quadrado que seja possível colocar numa certa linha. Outro que no pior caso pode ser iterado n-2 vezes e engloba a chamada de duas funções de complexidade $Θ(n^2)$, uma função onde a matriz é atualizada retirando um quadrado e contando uma opção e outra que contabiliza as colunas que estiverem vazias na matriz resultante da função anterior o que implica uma complexidade de $\Theta(n^3)$.

Loop1:
```
For n do
    condição;
end for
```

Loop2:
```
while n do
    chamadaFunção;
    chamadaFunção2;
end while
```

Assim, a função recursiva como é dependente do loop2 terá também uma complexidade de $\Theta(n^3)$.

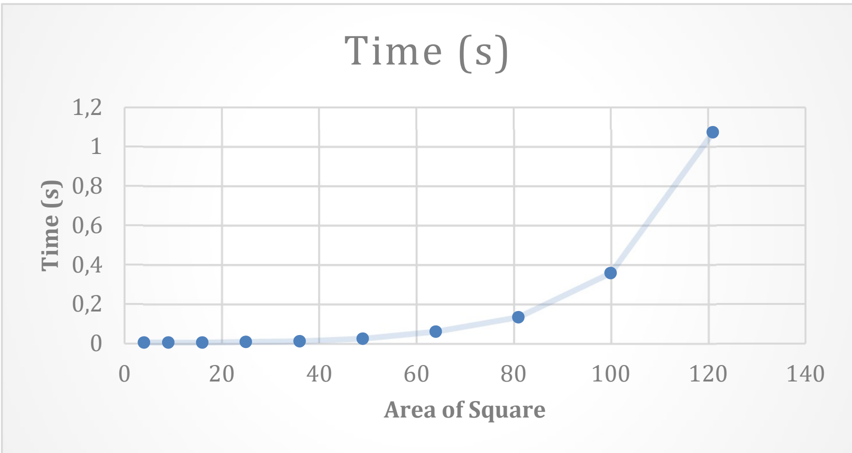
Por fim, é feito o output que consiste numa operação de complexidade $\Theta(1)$.

Concluindo, a solução ao problema apresenta complexidade $\Theta(n^3)$.

Avaliação Experimental dos Resultados

De forma a averiguar a complexidade real da solução ao problema, realizamos testes onde calculamos o tempo demorado a calcular o número de combinações de ladrilhos em quadrados perfeitos. Assim, os inputs do teste têm um formato constante e permitem obter uma cadeia de resultados de melhor compreensão. Os resultados dos testes foram os seguintes:

Area of Square	Time (s)
1	0,014
4	0,014
9	0,014
16	0,015
25	0,017
36	0,025
49	0,056
64	0,11
81	0,259
100	0,742



Analisando os resultados dos testes, é verificavel que os tempos de execução vão aumentando de uma forma nao linear, que se aproxima ao comportamento de uma função $y=n^3$. Assim, conclui-se que a solução encontrada para o problema na prática tem um comportamento que vai de encontro ao esperado durante a análise teórica, ou seja uma complexidade $\Theta(n^3)$.