Descritivo do algoritmo e análise Big-O

Algoritmo

O algoritmo busca qual a linha vertical traçada que intercepta menos tijolos em uma parede de tijolos e retorna o mínimo valor de tijolos interceptados. Tendo em vista o objetivo de realizar uma análise assintótica, foram apenas utilizados métodos síncronos, uma vez que utilizar métodos assíncronos não influencia o grau da curva de crescimento do tempo de execução.

Ele é composto por duas funções principais, a “GetClearRowPaths” e “FindOptimizedPath”. Seus comportamentos estão descritos abaixo

GetClearRowPaths

Sua função é calcular os caminhos abertos da linha de Tijolos. Ela percorre uma lista de Tijolos, somando o tamanho do tijolo atual com o anterior e salvando resultado em uma nova lista. Ele ignora o último tijolo, pôr o final dele será no final da parede. Com a posição do final de cada tijolo especificada, duas linhas com valores repetidos possuem um caminho livre, sem corta nenhum tijolo.

FindOptimizedPath

Sua função é descobrir a linha vertical que passar por todas linhas e que atinge o menor número de tijolos. Ela percorre uma “Linha perfeita” de Tijolos, onde todos tijolos possuem valor unitário. Ao percorrer, checa se cada lista de tijolos possui a posição atual da “Linha perfeita” de Tijolos, se lista possuir tal valor, o valor 1 é adicionado a uma variável “sumOfOpen”. Ao finalizar de percorrer as possíveis linhas de Tijolos, se “sumOfOpen” for maior que “maxSumOfOpen”, que representa e maior número de abertura encontrados, “maxSumOfOpen” adquire o valor de sumOfOpen. Por fim, retorna a altura da parede subtraído de “maxSumOfOpen”.

Análise Big-O

Com a finalidade de estipular o limite assintótico da performance do algoritmo, é preciso primeiro analisar as instruções contidas nas funções.

GetClearRowPaths/GetOptimizedLine

Nessas funções, a quantidade máxima de instruções relevantes realizadas pode ser descrita por f(L\*T), sendo que L é o número de linhas e T o máximo número de Tijolos. Para simplificar, é assumido que o número L é igual ao de T, temos então f(n2)

FindOptimizedPath

Na determinada função, a quantidade máxima de instruções relevantes realizadas pode ser descrita por f(L\*L\*T). Da mesma forma que nas funções anteriores, podemos assumir que L e T possuem o mesmo valor, o que resulta em uma equação f(n3).

Com base na análise as funções, temos duas equações que, somadas, indicam o número de execuções do algoritmo. Tendo isso em vista, assumindo um tempo unitário para cada execução, o tempo limite de execução do algoritmo pode ser descrito por O((n2) + (n3)), ou, simplificando O(n3), tendo que n é o resultado da soma do numero de linhas e do número máximo de tijolos dividido por 2.