

Relatório 1º projecto ASA 2022/2023

Grupo: AL093

Aluno(s): Duarte São José(103708) e João Maçãs (99970)

Descrição do Problema e da Solução

O problema que nos foi proposto passava descobrir de quantas maneiras diferentes seria possível ladrilhar com quadrados uma área definida sobre um retângulo com n linhas e m colunas e delimitada por um caminho em escada. Numa fase inicial pensamos que a única abordagem possível seria a de contar todas as soluções através de tentativa e erro. No entanto, para além da dificuldade que tivemos em não gerar soluções repetidas a solução não seria eficiente, pelo que desistimos da mesma.

Na nossa abordagem final ao problema começamos por identificar subproblemas do problema inicial, sendo cada um destes o resultado do preenchimento de um quadrado (que inclui sempre o quadrado de lado 1 mais acima da coluna c mais à direita possível) sem nunca criar uma escada que não pudesse ser representada por um array das suas colunas. Deste modo, cada problema dá origem a um conjunto de subproblemas correspondentes à remoção de quadrados do problema inicial. Para representar a escada, decidimos utilizar um array com as colunas dadas por input, de forma a utilizar o mínimo possível de memória. Ainda no âmbito da eficiência, a nossa solução faz uso de memoization utilizando uma HashTable na qual cada chave (um `int64_t` representativo da configuração de uma escada) corresponde ao número de soluções para essa mesma escada. Para contar o número de soluções, chamamos recursivamente todos os filhos de uma mesma escada.

Análise Teórica

Para simplificação da análise teórica, assumimos que:

- n corresponde ao número de linhas do tabuleiro.
- m corresponde ao número de colunas do tabuleiro.
- v corresponde ao vetor representante da escada do tabuleiro atual.
- c corresponde à coluna não preenchida mais à direita do tabuleiro atual.
- l corresponde à linha mais acima da coluna c do tabuleiro atual.
- q corresponde ao tamanho máximo de quadrados a colocar na posição (l, c)
- Leitura dos dados e criação da representação da escada (`readFile()`) - $O(n*m)$.
 - Ciclo a depender linearmente de n para ler o input. Logo $O(n)$.
 - No caso de $n > m$, fazemos uma transposição da escada com a função `transpose()` que possui um for loop dentro de outro sendo um deles dependente de m e o outro dependente de n . Logo $O(n*m)$.
- Verificar se a área a ladrilhar é 0 - $O(1)$.
 - Ver se um array está vazio. Logo $O(1)$.
- Análise do tabuleiro para encontrar o tamanho do maior quadrado que cabe na área a ladrilhar (`findLargestSquare()`) - $O(\min(n, m))$.
 - Ciclo for a depender linearmente do menor entre n e m . Logo $O(\min(n, m))$.
- Aplicação do algoritmo `countWaysRecursive()` para contar todas as maneiras possíveis de ladrilhar a área recebida no input - $O(n^2 * 2^n)$.
 - Transformação do array da escada para `int64_t` para ser possível dar hash, através de um ciclo a depender linearmente de n . $O(n)$

Relatório 1º projecto ASA 2022/2023

Grupo: AL093

Aluno(s): Duarte São José(103708) e João Maças (99970)

- Verificação da existência da escada na hash. $\sim O(1)$ (aproximadamente constante mas varia consoante o tamanho da hashtable)
- Procura da coluna c através da função `maxElement()`. $O(n)$
- Ciclo a depender linearmente de n, para encontrar a linha l. $O(n)$
- Ciclo a depender de q para construir todos os tabuleiros sem o quadrado de tamanho entre 1 e q sucessivamente retirado da pos (l,c). $O(n^2 * 2^n)$
 - Ciclo a depender linearmente de n para criar lst (cópia de v). $O(n)$
 - Remoção do quadrado de tamanho q do array lst. $O(q)$
 - Chamada recursiva da mesma função com a nova escada(2^n)
- Inserção do número de soluções para o vetor v, na hashtable. $O(1)$
- Apresentação dos dados. $O(1)$
 - Imprime o número de formas de ladrilhar o tabuleiro. $O(1)$.

Complexidade global da solução: $O(n^2 * 2^n)$

Avaliação Experimental dos Resultados

Como podemos ver pelos gráficos abaixo, o nosso trabalho teve uma evolução temporal muito próxima do esperado, apesar de crescer ligeiramente mais lentamente uma vez que a previsão teórica se trata de um estudo de pior caso.

Gráfico do tempo demorado em função da dimensão do input (considerando apenas quadrados totalmente preenchidos por simplificação)

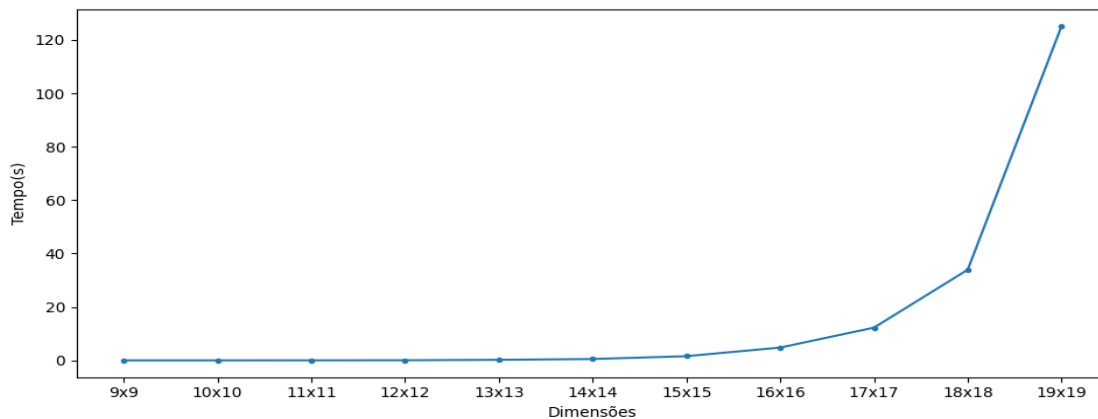


Gráfico da evolução esperada do tempo em função da dimensão do input

