Relatório 1º projecto ASA 2022/2023

Grupo: AL050

Aluno(s): André Filipe Silva Santos (103597)

Descrição do Problema e da Solução

Neste problema pretende-se encontrar o número de formas possíveis de preencher com quadrados uma grelha de n linhas e m colunas limitada por um caminho em escada.

Foi utilizada uma abordagem de programação dinâmica em que se resolve o problema dividindo-o em sub-problemas mais pequenos. Neste caso utilizando uma tabela de memoization é possível poupar tempo a resolver subproblemas que já tenham sido previamente calculados.

Utilizei como representação para o problema um array de inteiros em que cada valor i representa o tamanho da linha i da escada.

Após a leitura dos dados e a criação da array que representa o problema chama-se a função **calc_tilling_ways** que de forma recursiva irá resolver o problema da seguinte forma:

Primeiro utiliza-se uma função de hashing para dar hash ao array e verifica-se se a hash gerada já se encontra na tabela, isto é, se o problema já foi resolvido previamente. Caso o problema já tenha sido resolvido retorna-se o valor presente na tabela, caso contrário continua-se.

Calcula-se o tamanho da maior linha (degrau). Se este valor for menor ou igual a 1 estamos na presença do caso de paragem da função recursiva, isto é, em que não existe mais nada para ser calculado e retornasse 1, caso contrário continua-se.

Chegamos então ao núcleo da resolução do problema.

Encontra-se no array a primeira linha que tem o maior tamanho esta será a linha onde iremos trabalhar. Nessa linha calcula-se qual o tamanho do maior quadrado que pode ser removido, e depois remove-se nessa linha cada um dos quadrados começando no mais pequeno de tamanho 1 até ao maior utilizando uma função auxiliar que retorna o array resultante da sua remoção. É então chamada a própria função **calc_tilling_ways** com o array resultante da remoção e o resultado ao subproblema devolvido pela função é adicionado a um contador **"ways_of_tilling"**. Após correr este processo para todos os tamanhos (1 até ao máximo) é guardado na tabela o resultado da resolução deste problema em que a "key" da tabela é hash calculada previamente e o "value" é o número de soluções possíveis.

Por fim retornasse o número de soluções **"ways_of_tilling"**.

Análise Teórica

Assumindo uma grelha de n linhas e m colunas em que a escada não existe, isto é, a grelha é para preencher por completo.

- Leitura dos dados de entrada: $\Theta(n)$

Relatório 1º projecto ASA 2022/2023

Grupo: AL050

Aluno(s): André Filipe Silva Santos (103597)

- Hashing do array: O(n)
- Encontrar a maior linha: O(n)
- Encontrar a primeira linha com o maior valor: O(n)

Assumindo que k = min(n,m)

- Encontrar o tamanho do maior quadrado que pode ser removido na linha de maior valor: O(k)
- Remover quadrado: O(k)

Complexidade da solução:

T(n) = k * T(n-1) + O(n)

= k * [k * T(n-2) + O(n-1)] + O(n)

= k^k + k*O(n)

= k^k, sabendo que k = min(n,m)

Avaliação Experimental dos Resultados

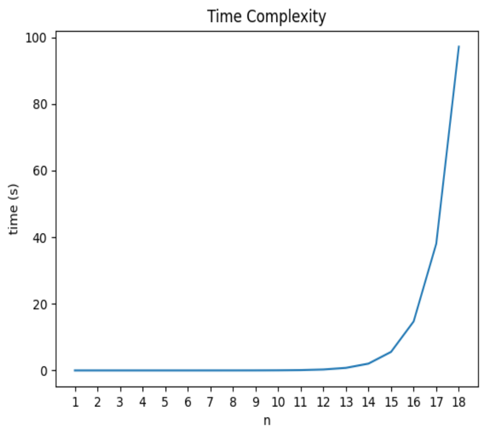
Pela minha análise, o pior caso do algoritmo dá-se quando a escada não existe, isto é, quando a grelha é para preencher por completo.

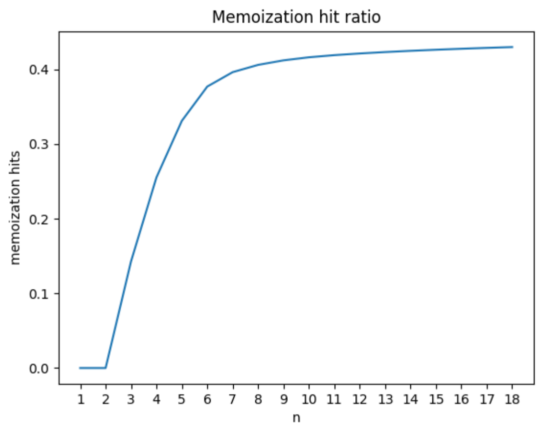
Os seguintes gráficos mostram em função da grelha nxn fornecida, (1) O tempo que o algoritmo demora, (2) O memoization hit ratio (número de vezes que o algoritmo retorna logo uma solução previamente armazenada).

Relatório 1º projecto ASA 2022/2023

Grupo: AL050

Aluno(s): André Filipe Silva Santos (103597)





Como se pode observar pelo gráfico (2) para grelhas cada vez maiores o memoization hit ratio está a tender para 43%.

Pela minha análise experimental acredito que a análise teórica esteja correta.