

Report for Programming Problem 1 - 2048 Team: 2015236199_2017267408

Student ID: 2015236199 Name: Tiago José Rodrigues Menezes

Student ID: 2017267408 Name: Pedro Carreiro Carvalho

Algorithm Description

Primeiro é colocado numa matriz o input fornecido, representando o estado do board inicial. Enquanto estes valores são lidos e inseridos na matriz é também calculado a soma de todos estes.

Após todos os dados estarem inseridos, é chamada uma função de análise aos valores ao qual é fornecida o valor da soma total dos mesmos e é verificado se esta soma é uma potência de 2. Caso seja, é possível resolver o problema, caso não seja, é impossível de resolver.

Para a execução de movimentos na board é utilizada uma aproximação em que a board é percorrida uma única vez e o resultado do movimento é escrito na matriz do movimento seguinte, não havendo a necessidade de realizar uma cópia da matriz que representa o estado atual do board. Esta função faz também a contagem de valores existentes após o movimento (*count*). Para tal é guardado o index no qual escrever o valor (*indextowrite*), um *int* temporário (*temp*) onde é guardado o valor a comparar e uma *flag* para garantir a correção da leitura dos elementos. Utilizando o movimento para a esquerda como exemplo: Para cada valor a *flag* é colocada a 0. A linha é percorrida e quando é lido o primeiro valor diferente de 0 este é guardado em *temp* e é incrementado *count*. O array continua a ser lido até encontrar o próximo valor diferente de 0. Caso este seja igual a *temp*, é escrito na matriz do movimento *temp*2*, incrementado o valor de *indextowrite*, *temp* é colocado a 0 e a *flag* é colocada a 1, de modo a que o valor lido não seja guardado em *temp*. Caso este seja diferente de *temp*, é escrito na matriz o valor de *temp*, este é igualado ao valor lido mais recentemente, aumentado de seguida o valor de *indextowrite* e *count*. Foi testado inicializar a matriz a 0 escrevendo os valores depois ou no final preencher o resto da matriz com 0. Optámos pela segunda opção porque não existe overlap na escrita e melhora a eficiência temporal da abordagem. Esta lógica é adaptada para os outros movimentos.

Após a análise pré recursão da board é chamada a função recursiva. Esta função consiste, sucintamente, em calcular todas as possibilidades existentes para o estado da board e verificar qual o número de movimentos mínimo necessário para que este fique apenas com um valor. Como *base case* temos o caso em que o número de valores existentes é apenas um e que o número de movimentos feitos até ao momento é menor do que a melhor solução encontrada até ao momento (*best*). Inicialmente este valor é igual ao número de movimentos mais um. Como condições de rejeição, existe a verificação se o número de movimentos feitos até ao momento é igual ao número de movimentos máximo ou se é maior ou igual *best*. Como esta abordagem seria

demasiado demorada foram implementados 3 *speed up tricks*. Um sendo a análise pré recursão referida anteriormente e os outros sendo referentes à exclusão de movimentos desnecessários. Estes movimentos considerados desnecessários são os movimentos para a esquerda/direita ou cima/baixo caso o movimento anterior para a esquerda/direita ou cima/baixo não tenha feito *merge* a nenhum valor, ou caso o movimento realizado deixe a board no mesmo estado. Para analisar se houve merge de valores a função recursiva recebe o número de valores existentes calculado pela função que realiza o movimento da board.

Existe também uma função para analisar se a board está exatamente igual ao que estava antes de realizar o movimento. Esta função percorre toda a matriz até encontrar uma diferença.

Data Structures

Para resolver este problema, é criada uma matriz tridimensional *auxvalues [move] [i] [j]*. Sendo que *move* representa o número de movimentos realizados até ao estado atual da board, representa o número da linha e *j* representa o número da coluna.

Correctness

Devido a termos obtido a classificação de 200 pontos iremos explicar o porquê da nossa abordagem ser correta. A abordagem é correta pois percorre todas as possibilidades existentes para o estado da board, excluindo apenas os ramos da árvore recursiva nos quais o movimento feito anteriormente é desnecessário. Caso o movimento deixe a board exatamente igual é, de forma óbvia, um movimento inútil, daí a exclusão deste e de todos os movimentos existentes após este movimento. A exclusão de movimentos laterais/verticais após um movimento lateral/vertical caso este não tenha realizado *merges* é também feita pois assim é evitado que sejam percorridos ramos da árvore recursiva que fariam movimentos laterais/verticais, até ao movimento ser *best* ou *MAX_MOVES*, cuja a única coisa que fariam seria mover as peças de um extremo para o outro sem a hipótese de encontrar um solução ou uma solução melhor. Excluímos deste corte apenas o primeiro movimento pois é possível que seja necessário alinhar os valores para fazer a junção dos mesmos.

Algorithm Analysis

A complexidade temporal deste algoritmo é $O((4^{\text{MAX_MOVES}}) * (N^2))$, sendo *N* a largura/altura da board, no *worst case*, pois pode ser necessário realizar os 4 movimentos por estado da board e porque os movimentos realizados na board são sempre quadráticos. Simplificada, a complexidade temporal é $O(4^{\text{MAX_MOVES}})$. Num caso ideal esta complexidade seria reduzida pois não seria necessário realizar todos os movimentos para cada estado da board, devido aos *speed up tricks* aplicados podendo ser $O(2^{\text{MAX_MOVES}})$, no *best case*.

A complexidade espacial deste algoritmo será $O((N^2) \cdot \text{MAX_MOVES})$, sendo N a largura/altura da board, pois poderá ter que guardar MAX_MOVES estados do tabuleiro. Existem casos em que não terão que ser guardados MAX_MOVES estados, pois pode ser encontrada uma solução com o valor menor que MAX_MOVES antes de atingir o mesmo e, nesse caso, não existe a necessidade de guardar os estados cujo movimento seja superior ao *best*.

References

Como referência foram apenas utilizados os slides fornecidos pelos professores e o apoio fornecido nas aulas práticas.