# **Relatório Trabalho 1 de ADA – Game of Beans**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| * **Ano Letivo** | * 2020/2021 | * **Semestre** | * 2 | * **Cadeira** | * ADA |
| * **Alunos** | * Gonçalo Martins Lourenço nº55780 | | | | |
| * Joana Soares Faria nº 55754 | | | | |

# Resolução do Problema

Definimos como sendo a função de nos dá a máxima pontuação da Jaba quando as pilhas em jogo começam na e acabam na , sendo que é a vez do jogador jogar.

Na definição desta função sentimos a necessidade de estabelecer duas funções auxiliares:

* que nos devolve a pontuação obtida por retirar pilhas do conjunto de pilhas que começa em e acaba em ;
* que nos devolve a melhor forma que o Pieton pode jogar, seguindo os seus critérios. Devolve-nos um array , de duas posições, cuja primeira posição nos diz o número de pilhas que o Pieton vai tirar da esquerda e a segunda nos diz o número de pilhas que o Pieton vai tirar da direita. (Tendo em consideração que se o Pieton retira pilhas da esquerda não irá tirar pilhas da direita e vice-versa).

Nesta função consideramos dois casos base, ambos que representam a situação em que há apenas uma pilha em jogo. Numa das situações é o Pieton a jogar, sendo que ele fica com o total da pontuação desse monte, dando à Jaba a pontuação de zero. Na outra situação é a Jaba a jogar, pelo que acumula a pontuação da pilha à sua pontuação final.

Nos casos gerais tratamos igualmente dos dois casos possíveis: é o Pieton a jogar ou é a Jaba a jogar, e consideramos sempre que pois se então já não haverá mais pilhas e o jogo estará terminado.

No caso de ser a Jaba a jogar a pontuação máxima que poderá obter será a melhor maneira possível de tirar pilhas ou da direita ou da esquerda, passando a jogada ao Pieton. Vamos então chamar recursivamente a função para saber a melhor forma de jogar nessa nova situação e acrescentamos à pontuação da Jaba o correspondente a tirar as pilhas da melhor forma.

No caso de ser o Pieton a jogar, então a pontuação que ele retirou não irá para a Jaba, sendo que a melhor pontuação da Jaba corresponderá a chamar recursivamente a função com menos as pilhas que o Pieton removeu, sendo que é a vez da Jaba jogar.

# Complexidade Temporal

Considerando as seguintes quantidades:

* , número de pilhas;
* , número de pilhas a considerar;
* , máximo de pilhas que é possível retirar que pode ser igual à profundidade ou menor se sobrarem menos pilhas ());
* , o número de pilhas a considerar retirar na jogada ();

Consideramos agora a complexidade temporal necessária para resolver o problema:

* Primeiramente efetuamos o preenchimento dos casos base na matriz de resultados que se traduz numa complexidade de ;
* Seguidamente, o tempo de preenchimento da tabela de resultados dos restantes valores é calculado por:
  + Começamos com um primeiro ciclo que representa o número de pilhas a considerar (). Este ciclo é efetuado vezes pelo que temos complexidade de
  + Aninhado neste primeiro ciclo temos outro ciclo que representa o ponto inicial onde começar a considerar a pilha, o (), sendo que executamos este ciclo vezes, obtendo uma complexidade de );
  + Finalmente, neste ciclo começamos por calcular a pontuação da Jaba, em que vamos determinar quantas pilhas vamos retirar. Para isso executamos um ciclo vezes, em que é o número máximo de pilhas que se pode retirar, e dentro desse ciclo somamos a pontuação de tirar as pilhas. (). Obtemos assim uma complexidade de , porque temos de considerar retirar as pilhas da esquerda ou da direita;
  + Acabamos por calcular a pontuação da Jaba se for uma jogada do Pieton e para isso temos de determinar quantas pilhas o Pieton vai retirar e por um raciocínio semelhante ao anterior obtemos novamente ;

Tomando todos estes aspetos em consideração obtemos a seguinte complexidade temporal:

e como , simplificando obtemos uma complexidade temporal de

# Complexidade Espacial

Considerando as seguintes quantidades:

* , número de pilhas;

A complexidade espacial do problema pode dividir-se entre as seguintes:

* Complexidade constante de duas variáveis (, ou número de colunas permitido retirar da pilha, e a correspondente ao nome do primeiro jogador, acrescendo mais as variáveis auxiliares declaradas nos diferentes métodos. Resulta assim em
* Complexidade espacial linear, das pilhas dadas pelo enunciado, ;
* Por fim, para guardar as pontuações da Jaba usamos uma matriz de três dimensões, por por , e assim sendo temos uma complexidade espacial de .

Concluindo, temos uma complexidade espacial de:

# Conclusões

Indiquem os pontos fortes e fracos da vossa solução, as alternativas estudadas ou que mereciam ser estudadas, possíveis melhoramentos, etc.

A nossa solução responde ao problema em tempo polinomial, o que é um benefício relativamente ao uso de uma função recursiva que teria chamadas repetidas, o que levaria a uma complexidade temporal.

Uma possível melhoria no algoritmo implementado seria a redução da complexidade espacial, usando apenas uma matriz bidimensional, representando cada célula da matriz como o melhor resultado da pontuação da Jaba numa ronda do jogo (considerando uma ronda uma jogada de cada jogador). Não optamos por esta solução porque o código produzido tinha algumas falhas e tinha uma compreensão um pouco mais difícil.

# Anexo – Código Main

# Anexo – Código Class GameOfBeans