

# Gustavo da Mota Ramos

## Resultado do Teste

### Questão 1

Você tem 8 moedas e todas tem o mesmo peso exceto uma, que é um pouco mais pesada que as demais. Você também possui uma balança que permite pesar duas pilhas de moedas para descobrir qual é a mais pesada (ou de mesmo peso). Qual é o mínimo número de combinações para serem feitas para descobrir qual a moeda mais pesada?

Possuo 8 moedas, 7 delas possuem o mesmo peso e uma possui o peso maior que as restantes. Como possuo uma balança capaz de medir pilhas de moedas, inicialmente coloco 4 empilhadas no lado esquerdo da balança e 4 empilhadas no lado direito da mesma balança, assim um dos lados terá o peso maior e consequentemente sei que nesta pilha está a moeda, dessa maneira consigo reduzir de 8 para 4 candidatas.

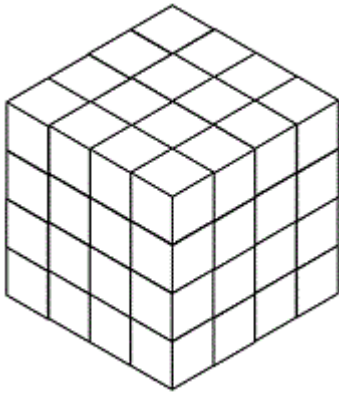
Para concluir, existem duas maneiras para localizar a moeda entre as 4 as quais chamarei de maneira 1 e maneira2, a maneira 1 possui um número fixo de passos que são 2, a maneira2 também possui 2 passos mas com uma diferença: na maneira dois existe 25% de chance de que a moeda saia já no primeiro passo reduzindo assim um passo.

Maneira 1: Supondo que eu tenha as moedas A,B,C,D ; Coloco as moedas A e B no lado esquerdo, C e D no lado direito da balança, um dos lados apontará um maior peso e isso ocorre pois a moeda mais pesada estará em um dos lados. Supondo que a pilha com A e B ficou mais pesada, então coloco A no lado esquerdo e B no lado direito, a balança irá mostrar a moeda mais pesada totalizando 3 medições ao todo.

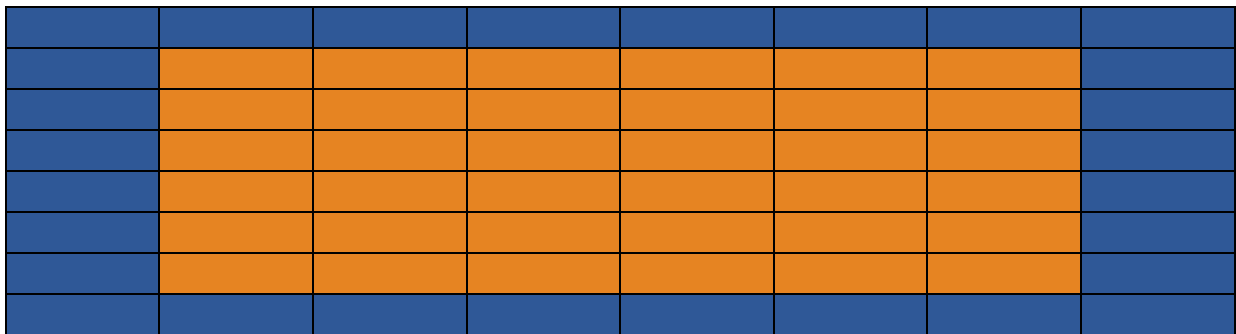
Maneira 2: Supondo que eu tenha as moedas A,B,C,D novamente; Escolho arbitrariamente duas entre elas e as coloco uma em cada lado da balança, neste caso, existe 25% de chance da moeda mais pesada já aparecer nesta primeira medição, o que a torna mais eficiente. Caso isso não ocorra (Neste caso as moedas apontariam o mesmo peso) isso diz que a moeda mais pesada é uma das duas outras não utilizadas, sendo assim fazendo mais uma medição entre essas não utilizadas é certeza que ela aparecerá. No melhor caso a moeda mais pesada apareceria em duas medições, e no máximo em 3 utilizando a maneira 2.

## Questão 2

Assuma um cubo semelhante ao do desenho abaixo, no entanto com 8 mini cubos em cada aresta. Se pintarmos todas as suas faces, quantos mini cubos terão pelo menos uma face pintada?



O cubo no total possui  $8 \times 8 \times 8$  mini cubos totalizando 512 mini cubos, ao pintar todas as faces teremos que somente o núcleo dele não terá contato com a tinta, supondo o cubo de lado:



Haverá um cubo interno de tamanho  $6 \times 6 \times 6$  ao qual não será pintado, se subtrairmos o total de cubos pela quantidade deste cubo interno teremos o complementar que é a quantidade de cubos pintados em pelo menos uma face.

Sendo assim o total de cubos é  $8 \times 8 \times 8 = 512$ , o cubo interno tem tamanho  $6 \times 6 \times 6 = 216$ , subtraindo o total pelo tamanho do interno:  $512 - 216 = 296$  cubos, ou seja, 296 cubos terão pelo menos uma face pintada.

## Questão 1

Dado um array de inteiros ordenados, como encontrar a posição de determinado valor?

Existem diversas maneiras de encontrar um valor em um array ordenado de tamanho  $n$  inteiro, porém duas se destacam, a primeira consiste em fazer uma busca exaustiva partindo do primeiro item na posição 0, ou seja `array[0]`, e prosseguir a busca até o último elemento de posição  $n-1$ , `array[n-1]`. O problema aqui é que o melhor caso é o qual o elemento procurado está na primeira posição, posição 0, e o pior caso na última posição, posição  $n-1$ , sendo assim pode não ser a melhor opção por ter uma complexidade maior do que outras opções ou no caso linear.

A segunda maneira, a qual prefiro utilizar, é fazendo uma busca binária no array, já que ele já está ordenado; Essa busca consiste em particionar o array na metade a cada iteração, de maneira que sempre se aproxime do valor, caso ele exista no array é claro.

Por exemplo: Para procurar 2 no array: `[1,2,3,4,5,6]`

Procuro o elemento central dele e verifico se é maior, se encontrei ou menor que o elemento que desejo, se for menor utilizo a parte da esquerda, caso contrário da direita, se é igual foi encontrado. Ao fazer isso restará: `[1,2,3]`. Faço o mesmo procedimento, o elemento central é 2, encontrei o elemento e sua posição é 1 (Em uma linguagem onde o array se inicia em zero)

## Questão 2

Escreva um código para reverter uma string.

O código foi escrito em java, possui duas classes, uma de teste unitário utilizando TDD e outra do código propriamente dito, está no repositório juntamente com este arquivo.