

PROGRAMAÇÃO COMPETITIVA - P1

Christian Junji Litzinger State
Filipe Gomes Arante de Souza
Vinicius Moraes de Jesus

Problem D: Counting Kangaroos is Fun

Problema: Qual a quantidade mínima de cangurus visíveis?

Utilizar dois indexadores, um começando na metade superior e outro na metade inferior.

No melhor dos casos, metade dos cangurus se esconde na outra metade.

Por que um canguru da primeira metade não é testado com um da primeira?

Por que um canguru da segunda metade não é testado com um da segunda metade?

Complexidade: Depende da implementação do algoritmo de ordenação do vetor, assumindo MergeSort, por conveniência ($O(n \lg(n))$).

PROBLEM D CODE

<https://github.com/gomesfilipe/prog-comp/blob/main/prova1/D/d.cpp>

Problem E: Potentiometers

Array de resistores, onde é possível:

- Consultar a resistência parcial de um segmento do array;
- Mudar o valor da resistência de algum resistor;

Como fazer uma quantidade muito grande dessas operações de um jeito eficiente?

Solução trivial não passa;

Ideia: árvore de segmentação!!!

1º Passo: construir a árvore de segmentação [$O(n)$];

2º Passo: efetuar as queries/updates [$O(\lg n)$ para cada consulta];

Complexidade total:

$$\underline{O(n)} + \text{qtd_queries} * O(\lg n) = O(n)$$

PROBLEM E CODE

<https://github.com/gomesfilipe/prog-comp/blob/main/prova1/E/e.cpp>

Problem F: Flavius Josephus Reloaded

O problema consiste em um ciclo e é possível aplicar o algoritmo da lebre e da tartaruga.

Complexidade $O(2N)$

- **N soldados numerados de 0 até $N - 1$.**
 - **para saber o próximo soldado e feito o seguinte cálculo:**
 - **$((a \cdot x^2) + b) \bmod N$**
 - **Soldados morre se seu número for calculado 2 vezes**
 - **Acaba quando o número de um soldado e calculado pela 3ª vez**
-

PROBLEM F CODE

<https://github.com/gomesfilipe/prog-comp/blob/main/prova1/F/f.py>