

PROCESSOS SELETIVOS (/MY_GAME) / 4: QUERO SER DEV 40+ (/SAGAS/4-QUERO-SER-DEV-40)
/ EXCLUSIVOS QUERO SER DEV 40+ (/SAGAS/4-QUERO-SER-DEV-40#CODE-WORLD-35)
/ CAIXA DA SORTE (/CHALLENGES/CAIXA-DA-SORTE)

Caixa da sorte

Aguardando...

Submeta sua solução através do Git para conquistar seus codepoints

Um determinado canal de TV tem um show bastante popular, pois permite que participantes liguem para o programa em um segmento específico com a oportunidade de ganhar prêmios em dinheiro. Esse segmento consiste de um jogo que envolve um conjunto de números pré-determinado colocado dentro de caixas (um por caixa). Um participante que consegue se conectar ao programa deve acertar em qual das caixas se encontra o maior número. Para isso, o participante dispõe de algumas informações:

No momento em que o programa começa, os números em todas as caixas são revelados. A partir daí, os números nas caixas são alterados, a cada minuto (e todos ao mesmo tempo), da seguinte forma:

- Um número em uma caixa de número ímpar é reduzido pela metade e esse valor resultante é somado ao valor da caixa ímpar seguinte. A última caixa de número ímpar deve ter metade de seu número transferido para a primeira caixa;
- Um número em uma caixa de número par é reduzido em metade de seu valor e esse valor resultante é somado ao valor da caixa de número par anterior. A primeira caixa de número par deve ter metade de seu valor transferido para a última caixa;

Você logo percebe que esse jogo pode ser traduzido em um algoritmo de forma que você sempre saiba qual a caixa que possui o número de maior valor.

Neste desafio, você deve implementar esse algoritmo que, dado o conjunto de números revelados no início do programa `numbers` e a quantidade de minutos passados até o momento em que você é atendido `minutes`, determina qual a caixa que possui o maior

número e qual o valor correspondente. Note que, em caso de empate, seu algoritmo deve retornar um array com todas as caixas que possuem o maior número.

Exemplo

Suponha uma chamada do algoritmo com os seguintes dados:

```
numbers = [2, 3, 4, 5, 6]
minutes = 1
```

A distribuição dos números das caixas, então, pode ser vista como:

Caixa	1	2	3	4	5
Número	2	3	4	5	6

Nesse caso, quando formos atendidos, 1 minuto terá se passado e, portanto, os números da caixa terão mudado. De acordo com as regras acima, todos os números são divididos em sua metade (usando uma divisão inteira) e, portanto, os números dentro da caixa se encontram da forma:

```
changed_numbers = [1, 1, 2, 2, 3]
```

Pois $2/2 == 1$, $3/2 == 1$, $4/2 == 2$, $5/2 == 2$ e $6/2 == 3$. Além desses números serem reduzidos à metade, agora precisamos somar os valores resultantes entre as caixas. Uma caixa de número ímpar tem seu valor somado à caixa ímpar seguinte e, daí, o número na caixa 1 é adicionado ao número da caixa 3, o número na caixa 3 é adicionado ao número da caixa 5 e o número na caixa 5 é adicionado ao número da caixa 1. Uma caixa de número par tem seu valor somado à caixa par anterior, então o número na caixa 2 é adicionado ao número da caixa 4 e o número na caixa 4 é adicionado ao número da caixa 2. Ao fazer essas operações, as caixas recebem os seguintes números:

```
resulting_numbers = [4,3,3,3,5]
```

A partir daí, a caixa com o maior número é a caixa 5 e seu valor também é 5.

Observação

Nesse desafio, considere que 'reduzir um número à sua metade' é o ato de realizar a divisão inteira desse número por 2. Ou seja, a "metade" de 5, dado por $5/2$, resulta em 2.

Instruções

Em seu terminal, com Git configurado e uma chave SSH cadastrada no CodeSaga, siga os passos abaixo:

Passo 1 - Clone o repositório

```
git clone git@git.codesaga.com.br:luizf/game-show
```

Passo 2 - Mude para o novo diretório

```
cd game-show
```

Passo 3 - Execute o script para configurar seu projeto

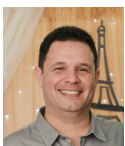
```
bin/setup
```

[PEDIR AJUDA \(/CHALLENGES/CAIXA-DA-SORTE/HELP_REQUESTS/NEW\)](/CHALLENGES/CAIXA-DA-SORTE/HELP_REQUESTS/NEW)

*(obrigatório) Comentário

Escreva aqui seu comentário

ENVIAR



Dimas Aparecido Alves (/users/dimas_alves) há aproximadamente 2 horas

Este desafio me lembrou de um filme muito antigo, o título: Retroceder nunca, Render-se jamais. Realmente, foi bem trabalhoso, dias pensando, calculando... Mas estamos somando ao nosso aprendizado. Sigamos para o alvo!!

[RESPONDER \(/COMMENTS/13795/REPLIES/NEW\)](/COMMENTS/13795/REPLIES/NEW)

👍 0 (/COMMENTS/13795/LIKE)



Luis Barufi (/users/luisbarufi) há 1 dia

Esse foi um dos mais difíceis, mas felizmente consegui finalizar!

[RESPONDER \(/COMMENTS/13663/REPLIES/NEW\)](/COMMENTS/13663/REPLIES/NEW)

👍 0 (/COMMENTS/13663/LIKE)



Ricardo Miranda de Novais (/users/ricardomiranda) há 1 dia

Desafio concluído

[RESPONDER \(/COMMENTS/13632/REPLIES/NEW\)](#)

0 (/COMMENTS/13632/LIKE)



Daniel Herszenhaut (/users/danhe) há 3 dias

Achei o erro no algoritmo

[RESPONDER \(/COMMENTS/13234/REPLIES/NEW\)](#)

1 (/COMMENTS/13234/LIKE)



Conrado Alceste Montienri Junior (/users/conradoalceste) há 3 dias

Daniel, quando eu tentei fazer na mão o procedimento indicado na orientação dada, não conseguia chegar ao resultado. Não sei se a descrição foi feita para nos fazer tentar chegar à lógica de como resolver o problema, mas a meu ver o enunciado está incompleto e confuso. Se você notar o resultado, ele não bate com a descrição do problema. Eu fiquei analisando e cheguei à seguinte conclusão: numbers = [2, 3, 4, 5, 6] minutes = 1 Portanto apenas uma rodada novo_numbers = [1, 1, 2, 2, 3] => nubers dividido por 2 resultado após a rodada: res = [4, 3, 3, 3, 5] Note que: res[0] = novo_numbers[4] + novo_numbers [0] res[1] = novo_numbers[3] + novo_numbers [1] res[2] = novo_numbers[0] + novo_numbers [2] Acho que no enunciado, entenda caixa por index. Os ímpares têm de pegar o anterior e os pares o posterior, e todos os valores divididos por 2, ao contrário do que dá a entender o enunciado. Foi fazendo dessa forma que consegui resolver o desafio.

[RESPONDER \(/COMMENTS/13232/REPLIES/NEW\)](#)

1 (/COMMENTS/13232/LIKE)



Daniel Herszenhaut (/users/danhe) há 3 dias

Oi Pessoal, mais um desafio que peço ajuda. Quando rodo o rspec tenho o seguinte retorno:

GameShow

- 1: Recebe 5 caixas com tempo de 1 minuto e retorna [[5,5]]
- 2: Recebe 5 caixas com tempo de 2 minutos e retorna [[1,4]]
- 3: Recebe 5 caixas com tempo de 4 minutos e retorna [[2, 2],[4, 2]]
- 4: Recebe 10 caixas com tempo de 6 minutos e retorna [[5, 49]]
- 5: Recebe 10 caixas com tempo de 6 minutos e retorna [[2, 53],[4, 53]] (FAILED - 1)
- 6: Recebe 12 caixas com tempo de 15 minutos e retorna [[9, 196]] (FAILED - 2)
- 7: Recebe 12 caixas com tempo de 30 minutos e retorna 6 caixas com número 186 (FAILED - 3)

Failures:

- 1) GameShow 5: Recebe 10 caixas com tempo de 6 minutos e retorna [[2, 53],[4, 53]]

Failure/Error: expect(GameShow.new.find_highest_number(numbers, minutes)).to eq ([[2, 53],[4, 53]])

expected: [[2, 53], [4, 53]]

got: [[2, 60], [4, 60], [6, 60], [8, 60], [10, 60]]

(compared using ==)

./spec/game_show_spec.rb:36:in `block (2 levels) in <top (required)>'

- 2) GameShow 6: Recebe 12 caixas com tempo de 15 minutos e retorna [[9, 196]]

Failure/Error: expect(GameShow.new.find_highest_number(numbers, minutes)).to eq ([[9, 196]])

expected: [[9, 196]]

got: [[2, 206], [12, 206]]

(compared using ==)

./spec/game_show_spec.rb:43:in `block (2 levels) in <top (required)>'

- 3) GameShow 7: Recebe 12 caixas com tempo de 30 minutos e retorna 6 caixas com número 186

Failure/Error: expect(GameShow.new.find_highest_number(numbers, minutes)).to eq ([[1, 186], [3, 186], [5, 186], [7, 186], [9, 186], [11, 186]])

expected: [[1, 186], [3, 186], [5, 186], [7, 186], [9, 186], [11, 186]]

got: [[2, 206], [12, 206]]

(compared using ==)

./spec/game_show_spec.rb:50:in `block (2 levels) in <top (required)>'

Finished in 0.02349 seconds (files took 0.45227 seconds to load)

7 examples, 3 failures

Failed examples:

rspec ./spec/game_show_spec.rb:32 # GameShow 5: Recebe 10 caixas com tempo de 6 minutos e retorna [[2, 53],[4, 53]]

rspec ./spec/game_show_spec.rb:39 # GameShow 6: Recebe 12 caixas com tempo de 15 minutos e retorna [[9, 196]]

rspec ./spec/game_show_spec.rb:46 # GameShow 7: Recebe 12 caixas com tempo de 30 minutos e retorna 6 caixas com número 186``

O que não entendo é que os exemplos 4 e 5 são praticamente iguais, mesmos minutos e array de mesmo tamanho. Não consigo imaginar onde pode estar o problema no meu código.

Alguma ideia? Se precisar do código me fala que eu coloco em um comentário.

RESPONDER (/COMMENTS/13218/REPLIES/NEW)

👍 0 (/COMMENTS/13218/LIKE)



André Kanamura (/users/akaninja) ✓ há 3 dias

Para tirar dúvidas sobre seu código, sempre abra pedidos de ajuda.

👍 0 (/COMMENTS/13358/LIKE)



Daniel Herszenhaut (/users/danhe) há 2 dias

Oi André, eu sei. Estava exausto e acabei me confundindo. Peço desculpas.

Não sei se seria possível apagar esse meu comentário.

👍 0 (/COMMENTS/13392/LIKE)

(<http://campuscode.com.br/>)

ENTRE EM CONTATO

codesaga@campuscode.com.br

Como funciona (/pages/como-funciona)

Feito por Campus Code (<https://campuscode.com.br>)

Termos de Uso e Política de Privacidade (/terms)