

# Tutorial: Adição e Subtração

Daniel Saad Nogueira Nunes

Uma forma de resolver esse problema é usando busca completa. Pode se usar um vetor de bits para representar todos os padrões no intervalo  $[0, 2^n - 1]$ . Para cada padrão  $X = x_1x_2 \dots x_{n-1}$  fazemos o seguinte: se  $x_i$  igual a 1 então, o  $i$ -ésimo sinal corresponde a +, caso contrário, a um -. Em seguida, realizamos as operações correspondentes ao padrão e verificamos se o resultado é igual a  $m$ . Se for o caso, a resposta é SIM. Se nenhum padrão garantir o resultado, a resposta é NAO. A complexidade da solução é  $\Theta(n2^n)$ .

# Tutorial: Conectado

Jeremias Moreira Gomes

É possível resolver utilizando 3 pilhas (inicial, objetivo e reserva), para tentar simular o comportamento esperado de uma organização correta. Inicialmente verifica-se se há alguma cor no topo da pilha reserva e ela é igual à próxima do objetivo, senão verifica-se se a cor no topo da inicial é igual à do objetivo ou, então, guarda-se a cor de inicial em reserva. Se alguma dessas três operações não puder ser feita, responde-se com “Defeito de fabrica!”

```

pilha -> inicial, objetivo, reserva
while objetivo possuir elementos:
    if reserva.topo() == objetivo.topo():
        reserva.remove()
        objetivo.remove()
        'R'
    else if inicial.topo() == objetivo.topo():
        inicial.remove()
        objetivo.remove()
        'M'
    elif inicial and objetivo:
        reserva.insere(inicial.remove())
        'G'
    else:
        imprime “Defeito de Fabrica!”
```

# Tutorial: Daileon

Daniel Saad Nogueira Nunes

Este problema pode ser reduzido a um problema de conectividade em grafos simples.

Primeiro, monta-se a lista de adjacências entre os receptores, Jaspion e Daleon. Dois nós possuem arestas entre si se a distância entre os centros da circunferência for menor, ou igual, à soma dos raios.

Em seguida, basta realizar uma busca em profundidade, ou em largura, para identificar se Jaspion alcança Daleon.

**Complexidade:** No pior caso, gasta-se tempo  $\Theta(n^2)$  para montar o grafo e tempo  $O(n^2)$  no percurso. Portanto, o algoritmo leva tempo  $\Theta(n^2)$ .

# Tutorial: Feijão, Feijão

Guilherme Ramos

Basta ler a entrada e verificar se há ou não a palavra “feijao” para escolher a resposta adequada.

# Tutorial: Igreja Invisível

Guilherme Ramos

A solução passa por armazenar as transformações possíveis e depois avaliar cada pedido. Se a transformação for possível, basta avançar por cada letra do resultado esperado e ajustá-la conforme a caixa do caractere relacionado na palavra de origem, ou com a do último caractere utilizado, se for o caso.

# Tutorial: Jogo da Velha

Daniel Saad Nogueira Nunes

As regras do jogo são simples. Basta verificar cada linha, coluna e diagonal. Se houver três símbolos consecutivos, o ganhador é o detentor daquele símbolo. Caso contrário, a resposta deve ser **empate**.