Lario & Muigi Adventures

Competição de Programação

Soluções

semana_
_de_informática

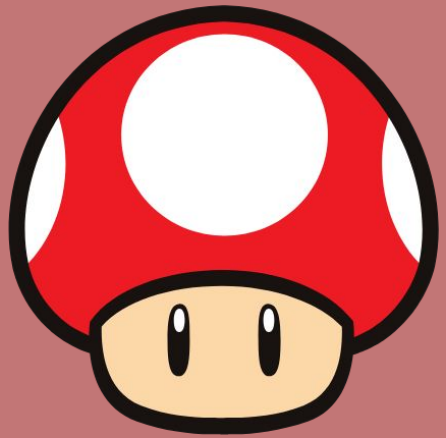
Lario and Mushrooms

Neste problema basta verificar em que categoria é que o inteiro fornecido se encaixa. Podemos alcançar isto com uma série de *if-else statements*.

Tema: Ad Hoc

Dificuldade: Fácil

Complexidade temporal: $O(1)$



Smash Browser

Neste problema temos de determinar se o Lario consegue vencer o Browser usando as C moedas. Não há qualquer restrição quanto ao gasto das moedas, por isso devemos ser *greedy* e gastar o maior número de moedas possível.

Tema: Complete Search, Greedy

Dificuldade: Médio -

Complexidade temporal: $O(C)$

Smash Browser

Como C é no máximo 10^5 , podemos explorar todas as possibilidades quanto ao uso das moedas: X moedas para vida e $C - X$ para dano.

Tema: Complete Search, Greedy

Dificuldade: Médio -

Complexidade temporal: $O(C)$

Smash Browser

Para cada iteração calcula-se a vida e dano com que o Lario fica. Podemos depois calcular o número de *hits* que cada personagem precisa para derrotar o outro.

Tema: Complete Search, Greedy

Dificuldade: Médio -

Complexidade temporal: $O(C)$

Smash Browser

Os *hits* são calculados dividindo a vida do personagem atacado pelo dano do personagem que ataca. Se a divisão não for inteira soma-se 1.

Tema: Complete Search, Greedy

Dificuldade: Médio -

Complexidade temporal: $O(C)$

Smash Browser

Se nalguma das iterações os *hits* do Lario forem menores ou iguais aos *hits* do Browser, Lario ganha, senão perde.

Tema: Complete Search, Greedy

Dificuldade: Médio -

Complexidade temporal: $O(C)$

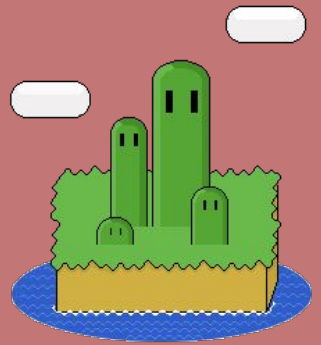
Squashing Mega Doombas

Observação chave: Se escolhermos usar uma montanha com altura H, podemos “passar por cima” de todas as próximas montanhas cuja altura é menor que H.

Tema: Greedy, Stacks

Dificuldade: Médio

Complexidade temporal: $O(N)$



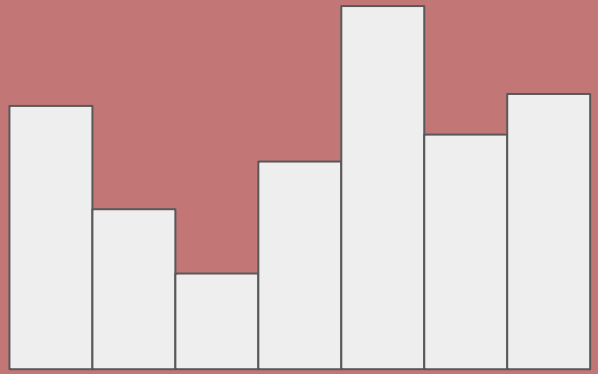
Squashing Mega Doombas

Assim podemos, da esquerda para a direita, escolher a montanha atual para ser um dos extremos da ponte, e iterar até à próxima montanha de altura maior ou igual, guardando a maior diferença de altura vista.

Tema: Greedy, Stacks

Dificuldade: Médio

Complexidade temporal: $O(N)$



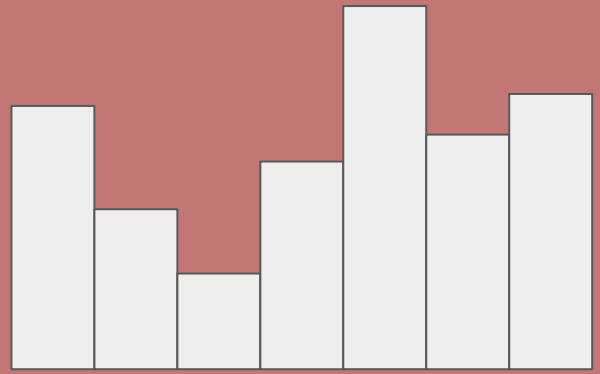
Squashing Mega Doombas

Contudo, para cada montanha que pode ser um “extremo”, temos de considerar construir a ponte para a direita ou para a esquerda!

Tema: Greedy, Stacks

Dificuldade: Médio

Complexidade temporal: $O(N)$



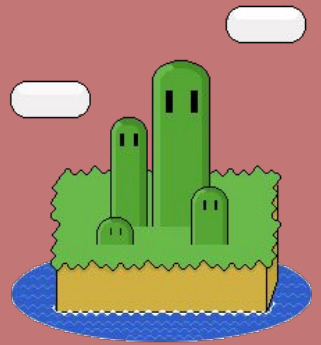
Squashing Mega Doombas

Para resolver este problema basta repetir o mesmo processo da esquerda para a direita e obter a máxima diferença de altura entre os dois processos.

Tema: Greedy, Stacks

Dificuldade: Médio

Complexidade temporal: $O(N)$



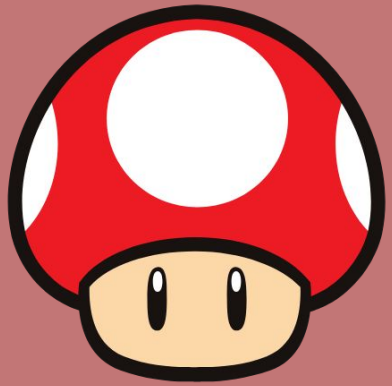
Mushroom Baskets

Havendo N cogumelos, existem 2^N cestos possíveis. Cada cogumelo pode pertencer a exatamente 2^{N-1} cestos! Com esta informação, podemos calcular o peso total de todos os cestos possíveis em $O(N)$.

Tema: Complete Search, Mathematics

Dificuldade: Médio +

Complexidade temporal: $O({}^{N}C_{4})$



Mushroom Baskets

Há também outra observação importante: dado que o mínimo de peso de um cogumelo são 50 gramas, são muitos menos os cestos com peso menor que 200 gramas do que ao contrário!

Tema: Complete Search, Mathematics

Dificuldade: Médio +

Complexidade temporal: $O(^{N}C_{4})$

Mushroom Baskets

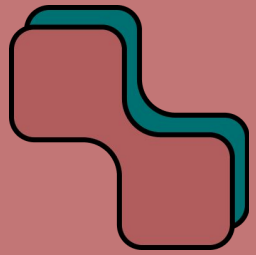
Assim devemos recursivamente (e com uso de *pruning*) calcular todos os cestos com peso menor do que 200 gramas. Depois basta subtrair este peso ao total e dar a resposta.

Tema: Complete Search, Mathematics

Dificuldade: Médio +

Complexidade temporal: $O(^{N}C_{4})$

Muigi’s Vacuum Supplier



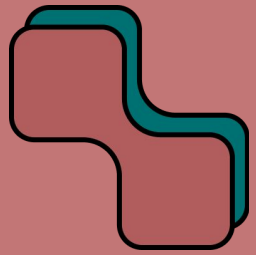
Observação chave: existindo apenas duas linhas de N colunas, o próximo espaço sujo ficará sempre ou na mesma coluna ou numa coluna à direita de onde se está atualmente.

Tema: Dynamic Programming

Dificuldade: Difícil

Complexidade temporal: $O(N)$

Muigi’s Vacuum Supplier



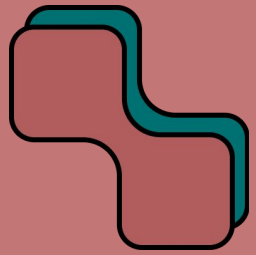
Podemos usar Dynamic Programming para resolver este exercício. Ao chegar a uma certa coluna vamos assumir que todos os espaços que estão nas colunas atrás estão limpos!

Tema: Dynamic Programming

Dificuldade: Difícil

Complexidade temporal: $O(N)$

Muigi’s Vacuum Supplier



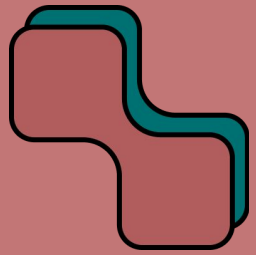
Vamos representar o nosso estado com três variáveis: a linha e coluna em que estamos e se o espaço na mesma coluna mas na outra linha vai ser deixado limpo ou não. Existem três hipóteses para passar para a próxima coluna:

Tema: Dynamic Programming

Dificuldade: Difícil

Complexidade temporal: $O(N)$

Muigi’s Vacuum Supplier

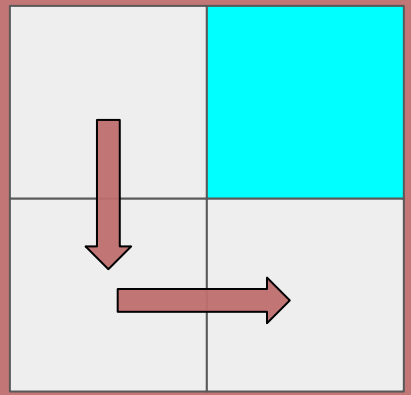


Vamos representar o nosso estado com três variáveis: a linha e coluna em que estamos e se o espaço na mesma coluna mas na outra linha vai ser deixado limpo ou não. Existem três hipóteses para passar para a próxima coluna:

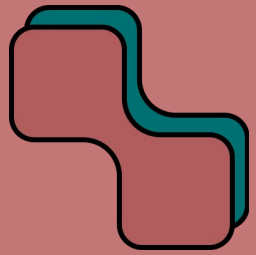
Tema: Dynamic Programming

Dificuldade: Difícil

Complexidade temporal: $O(N)$



Muigi’s Vacuum Supplier

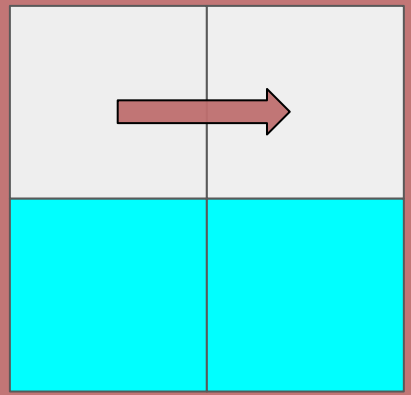


Vamos representar o nosso estado com três variáveis: a linha e coluna em que estamos e se o espaço na mesma coluna mas na outra linha vai ser deixado limpo ou não. Existem três hipóteses para passar para a próxima coluna:

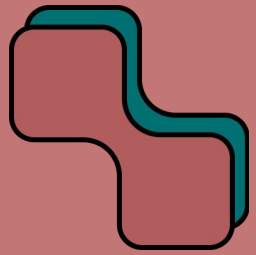
Tema: Dynamic Programming

Dificuldade: Difícil

Complexidade temporal: $O(N)$



Muigi’s Vacuum Supplier

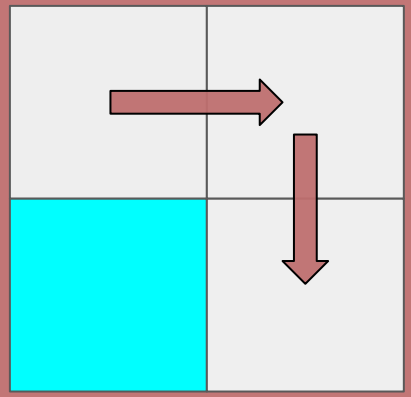


Vamos representar o nosso estado com três variáveis: a linha e coluna em que estamos e se o espaço na mesma coluna mas na outra linha vai ser deixado limpo ou não. Existem três hipóteses para passar para a próxima coluna:

Tema: Dynamic Programming

Dificuldade: Difícil

Complexidade temporal: $O(N)$



Lario and the Colliding Doombas

Este problema, apesar de parecer complexo, tem um “truque” que permite resolvê-lo de uma maneira muito simples!

Tema: Ad Hoc

Dificuldade: Tricky!

Complexidade temporal: $O(N)$

Lario and the Colliding Doombas

Vamos imaginar dois doombas prestes a colidir, representados abaixo:

Tema: Ad Hoc

Dificuldade: Tricky!

Complexidade temporal: *O*(N)

Lario and the Colliding Doombas

Vamos imaginar dois doombas prestes a colidir, representados abaixo:

Tema: Ad Hoc

Dificuldade: Tricky!

Complexidade temporal: $O(N)$

Lario and the Colliding Doombas

Vamos imaginar dois doombas prestes a colidir, representados abaixo:

Tema: Ad Hoc

Dificuldade: Tricky!

Complexidade temporal: $O(N)$

Lario and the Colliding Doombas

Vamos imaginar dois doombas prestes a colidir, representados abaixo:

Tema: Ad Hoc

Dificuldade: Tricky!

Complexidade temporal: $O(N)$

Lario and the Colliding Doombas

Em vez de pensarmos que os doombas chocam uns com os outros, podemos pensar que simplesmente passam uns pelos outros sem se tocarem.

Tema: Ad Hoc

Dificuldade: Tricky!

Complexidade temporal: $O(N)$

Lario and the Colliding Doombas

Assim precisamos apenas de ver a máxima distância entre a coordenada inicial de cada doomba e a ponta da plataforma para o qual ele está direcionado!

Tema: Ad Hoc

Dificuldade: Tricky!

Complexidade temporal: $O(N)$