Approach: Topological sort + layer BFS

$r_0 \; r_1 \; r_2 \; r_3 \; r_4 \; \cdots \; r_{i-1} \; r_i \; r_{i+1} \; \cdots \; r_{n-1}$

$c_0 \; c_1 \; c_2 \; c_3 \; \cdots \; c_{i-1} \; c_i \; c_{i+1} \; \cdots \; c_{n-1}$

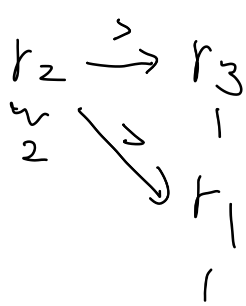
→ Goal: minimize $\sum_{i=0}^{n-1} c_i$

Constraints:

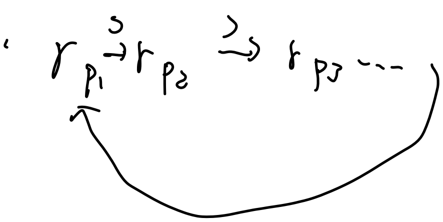
$c_i \geq 1$

and $\begin{cases} \text{If } r_i > r_{i-1}: & c_i > c_{i-1} \\ \text{If } r_i > r_{i+1}: & c_i > c_{i+1} \end{cases}$

$r_0 \; r_1 \; r_2 \; r_3 \; r_4$ acyclic



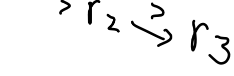
$\because$



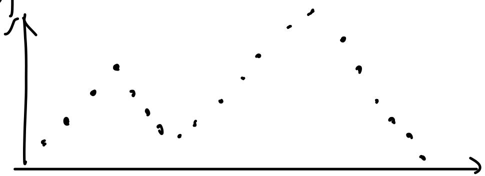
does not exist

$r_0 \; r_1 \; r_2 \; r_3$

$= 0 \; 2 \; 1 \; 0$



Appoarch: forward, backward DP with some derivations



$r_0 \quad r_1 \quad r_2 \quad r_3 \quad r_4$

$<$

12341111234567

Case 1 : $r_{i-1} < r_i$ and $r_i > r_{i+1} \Rightarrow C_i = \max(C_{i-1}+1, C_{i+1}+1)$

depends on left and right

Case 2 : $r_{i-1} < r_i$ and $r_i \le r_{i+1} \Rightarrow C_i = C_{i-1}+1$

→ depends on left

Case 3 : $r_{i-1} \ge r_i$ and $r_i > r_{i+1} \Rightarrow C_i = C_{i+1}+1$

→ depends on right

Case 4 : $r_{i-1} \ge r_i$ and $r_i \le r_{i-1} \Rightarrow C_i = 1$

→ do not depends on either left or right

Observe that:

When we see a $r_{i-1} < r_i$ (NOTE: we don't know r_i r_{i+1} relation)

Case ①: $r_{i-2} < r_{i-1} < r_i \Rightarrow C_{i-1} = C_{i-2}+1$

→ depends on its left

Case ②: $r_{i-2} \ge r_{i-1} < r_i \Rightarrow C_{i-1} = 1$

→ does not depend on its left

The first time we see a $r_{i-1} < r_i$

it must be like.

$$\ldots\ldots r_{i-2} \geq \underset{c_{i-1}=1}{r_{i-1}} < r_i$$

You can see $c_{i-1} = 1$, and we can put

$c_i \leftarrow c_{i-1} + 1$ even know we don't know

$r_i \,?\, r_{i+1}$ relation yet

Note that $r_0 \sim r_{i-3}$ will not affect

c_{i-1}'s value!

However, as you can see

$$r_{i-2} \geq r_{i-1} < r_i < r_{i+1} > r_{i+2}$$

$$c_{i-1} = 1 \quad c_i = 2 \quad c_{i+1} = 3$$

This is partially correct

because we need

$\max(c_i, c_{i+2})$!

but we alreay have "half"
That is c_i !
That is why we need
to do it backward again

Claim1 : every time we see a $r_{i-1} < r_i$
c_{i-1} must has the true value :

Base case :

The first time we see

$$\geq \cdots \geq r_{i-2} \geq r_{i-1} < r_i$$

$$\downarrow$$

$$c_{i-1} = 1$$

$\leftarrow$ correct !

Induction hypothesis: Assume it works when an array contains k $(r_{i-1} < r_i)$ pairs

Induction step: $k+1$ pairs of $(r_{i-1} < r_i)$

Suppose $r_{i-1} < r_i$ is the last pair

what we are interested

Case 1: $r_{i-2} < \boxed{r_{i-1} < r_i} \geq \cdots$

In this case, by our induction hypothesis, C_{i-2} contains the true value

and $\because \underbrace{r_{i-2} < r_{i-1} < r_i}_{\hookrightarrow \text{ Case 2 above}}$, $C_{i-1} = C_{i-2} + 1$ is true too

Case 2: $r_{i-2} \geq \boxed{r_{i-1} < r_i} \geq \cdots$

$C_{i-1} = 1$ is our initial value and it happens to be the true value

by claim 1:

after a forward dp,

$$\begin{array}{ccccccc} r_0 & r_1 & r_2 & r_3 & \cdots & & r_{n-1} \\ C_0 & C_1 & C_2 & C_3 & & & C_{n-1} \end{array}$$

Case 1: $r_{i-1} < r_i \leq r_{i-1}$:

C_i will be placed with a correct value

Case 2: $r_{i-1} < r_i > r_{i-1}$

c_i will be placed with $c_{i-1}+1$
which is half true

Case 3: otherwise ($r_{i-1} \geq r_i$)

c_i is placed as 1

Combine a forward and backward dp

$r_0 \; r_1 \; r_2 \; \cdots \; r_{n-1}$

forward $f_0 \; f_1 \; f_2 \; f_i \; f_{n-1}$

backward $b_0 \; b_1 \; b_2 \; b_i \; b_{n-1}$

true: $c_i = \max(f_i, b_i)$

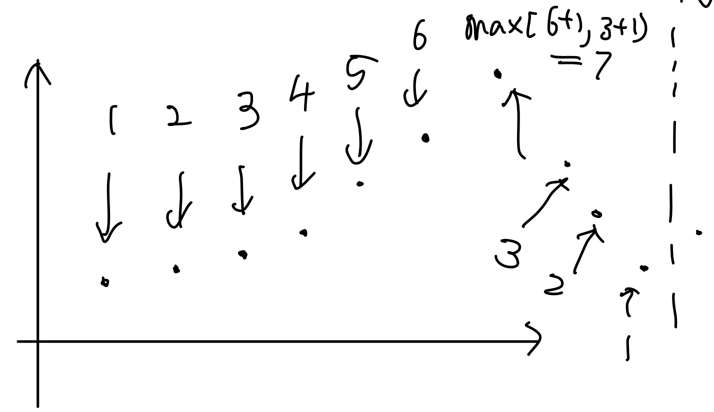
#

Approach 3: O(1) space

Observe that:

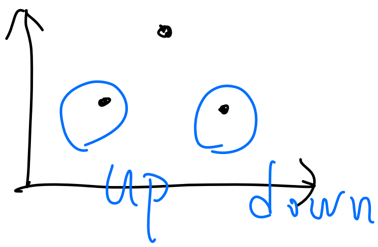
1. Local minimum must be $c_i = 1$

   (where $r_{i-1} \geq r_i \leq r_{i+1}$)

2. 

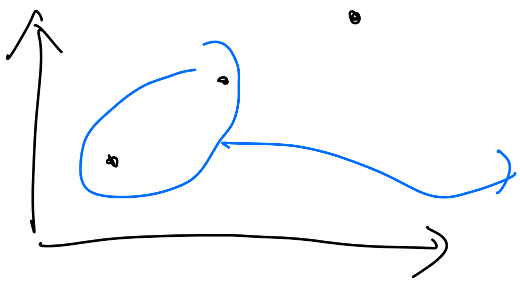
$\Rightarrow$ This tell us that after we finish a "mountain", we will know how many candies we need

Case 1: $r_{i-1} < r_i$ and $r_i > r_{i+1}$



$c_i = \max(c_{i-1}, c_{i+1}) + 1$

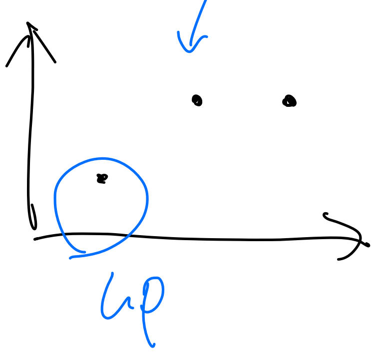
Case 2: $r_{i-1} < r_i$ and $r_i < r_{i+1}$



count as up

Case 3: $r_{i-1} < r_i$ and $r_i = r_{i+1}$

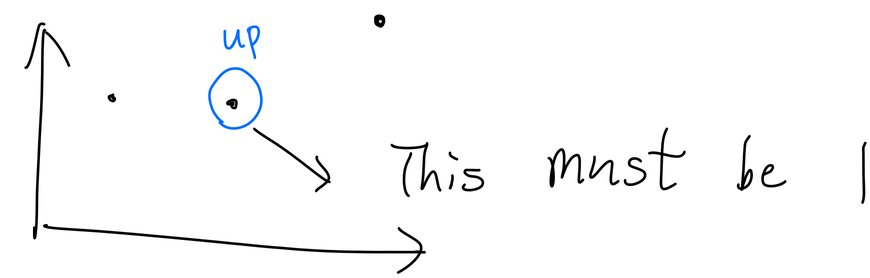
this peak's value is decided



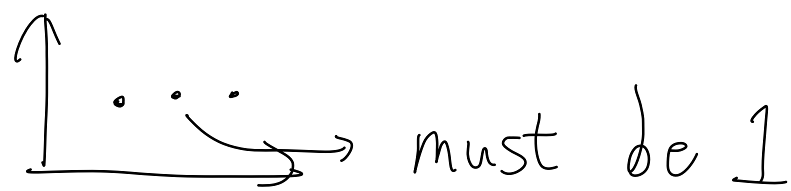
$\Rightarrow$ In this case, when we see $r_i = r_{i+1}$, we know c_i only depends on c_{i-1}

$\Rightarrow$ Compute

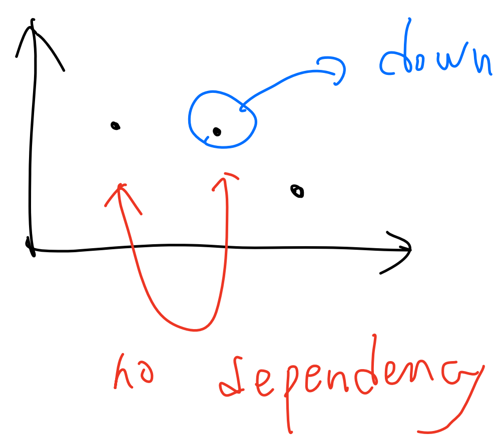
Case 4: $r_{i-1} = r_i$ and $r_i < r_{i+1}$



case 5: $r_{i-1} = r_i$ and $r_i = r_{i+1}$

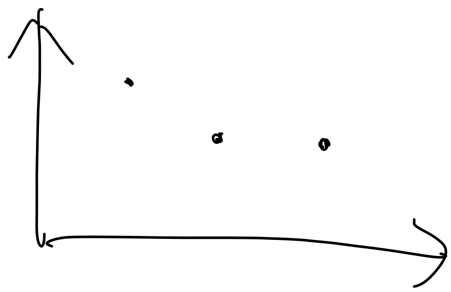


Case 6: $r_{i-1} = r_i$ and $r_i > r_{i+1}$



Case 7: $r_{i-1} > r_i$ and $r_i < r_{i+1}$

Case 8: $r_{i-1} > r_i$ and $r_i = r_{i+1}$



Case 9: $r_{i-1} > r_i$ and $r_i > r_{i+1}$