

Have you heard the "binary indexed tree"?

- <https://leetcode.com/problems/range-sum-query-mutable/>
- <http://www.csie.ntnu.edu.tw/~u91029/Sequence.html#8>

More hints in the next page ↓.

0							
4				0			
2		2		0		0	
1	1	1	1	1	0	1	0

Figure 1: The binary indexed tree

The upper level is:

- Addition of the lower 2 levels if they are both non-zero.
- 0, otherwise.

More hints in the next page ↓.

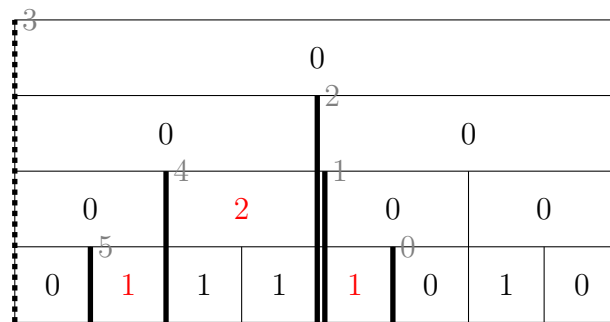


Figure 2: The binary indexed tree

Check	Result	Length	Loop invariant
Try to add 1 to align to 2	OK	1	Align to 2
Try to add 2 to align to 4	Already	1	Align to 4
Try to add 4 to align to 8	Fail	1	Align to 4, less than 4 more 1's
Try to add 2	OK	3	Less than 2 more 1's
Try to add 1	OK	4	Less than 1 more 1's (done)

Table 1: The Algorithm

We traverse the tree bottom-up then top-down. This algorithm can handle at most $(2^h - 1)$ 1's where h is the height of the tree. So you have to choose $h = 9$ to handle $k = 500$.

Note that we also illustrate the "OK", "Already" and initial state by bold black lines while "Fail" is dotted bold lines.

One more example in the next page ↓.

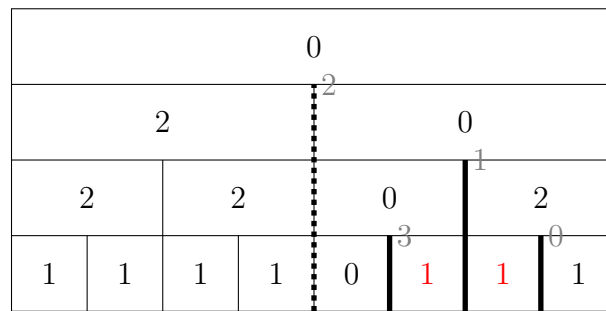


Figure 3: The binary indexed tree

Check	Result	Length	Loop invariant
Try to add 1 to align to 2	OK	1	Align to 2
Try to add 2 to align to 4	Fail	1	Align to 2, less than 2 more 1's
Try to add 1	OK	2	Less than 1 more 1's (done)

Table 2: The Algorithm

In this example we "Fail" at the lower level.

Sadly I have no more hint for you.