

# Analysis: Spectating Villages

## Test set 1 (Visible)

Since there are very few villages for this set, we can calculate the sum of beauty values of illuminated villages for every possible set of villages we choose to build lighthouses in and then take the maximum among these.

For each such set, we have to find which villages will be illuminated. So, for each village, we see if it is in the set or if any of its neighbours are in the set and add its beauty value accordingly.

Since there will be  $2^V$  such sets and for each such set we are taking  $O(V)$  time to figure out what is the corresponding total beauty value for the set, the overall complexity of this approach is  $O(V2^V)$

## Test set 2 (Hidden)

The the graph formed by taking the villages as nodes and roads as edges is a tree. Let's root this tree at node number 1.

Now, let us define a function  $\text{maxBeauty}(K, P, Q)$  which represents the maximum beauty value we can obtain from nodes in the subtree of node  $K$  (including itself) such that  $P$  is a boolean indicating whether the parent node of  $K$  has a lighthouse and  $Q$  is a boolean indicating whether  $K$  itself has a lighthouse. The solution to our problem is simply  $\max(\text{maxBeauty}(1, 0, 1), \text{maxBeauty}(1, 0, 0))$ . We consider a few cases to evaluate  $\text{maxBeauty}(K, P, Q)$ .

If  $Q = 1$ , then we have a lighthouse placed at  $K$ . Which means all of its children will be illuminated irrespective of them having a lighthouse or not. Therefore, in this case,  $\text{maxBeauty}(K, P, Q) = B_K + \text{sum of } \max(\text{maxBeauty}(C, 1, 0), \text{maxBeauty}(C, 1, 1)) \text{ for all children } C \text{ of node } K$ .

If  $Q = 0$  and  $P = 1$ , then irrespective of whatever we choose for the children of  $K$ , they are not going to receive light from  $K$  but  $K$  itself is going to be illuminated. Therefore, in this case,  $\text{maxBeauty}(K, P, Q) = B_K + \text{sum of } \max(\text{maxBeauty}(C, 0, 0), \text{maxBeauty}(C, 0, 1)) \text{ for all children } C \text{ of node } K$ .

Else, we have  $Q = 0$  and  $P = 0$ . This means that the children of  $K$  are not going to receive light from  $K$  but the illumination of  $K$  depends on whether we place a lighthouse in at least one of the children of  $K$ .

This case can be handled using a dynamic programming approach. We define  $\text{dp}[i][j]$  as maximum sum of beauty values of all illuminated nodes in the first  $i$  subtrees of node  $K$  such that, if  $j = 0$  then we have not placed a lighthouse in any of the first  $i$  children of  $K$  and if  $j = 1$  then there is at least one node among the first  $i$  children of  $K$  with a lighthouse.

Therefore,  $\text{dp}[i][0] = \text{dp}[i-1][0] + \text{maxBeauty}(C_i, 0, 0)$  and

$\text{dp}[i][1] = \max(\text{dp}[i-1][1] + \max(\text{maxBeauty}(C_i, 0, 0), \text{maxBeauty}(C_i, 0, 1)), \text{dp}[i-1][0] + \text{maxBeauty}(C_i, 0, 1))$ .

(Here,  $C_i$  represents the  $i$ th child of  $K$ )

Finally, for this case,  $\text{maxBeauty}(K, P, Q) = \max(\text{dp}[M][1] + B_K, \text{dp}[M][0])$  where  $M$  is the total number of children of  $K$ .

Since for every node  $K$  we can get  $\text{maxBeauty}(K, P, Q)$  for all values of  $P$  and  $Q$  in  $O(M)$  time, overall complexity of this approach is  $O(N)$ .