Question: https://leetcode.com/problems/minimum-time-to-make-rope-colorful/

So the logic is very simple, keep track if a certain colour i is present adjacently more than once for n times then except the one with smallest value remove the rest n-1 items.

We achieve this by using stack, i.e., we check if top of stack equal to present element, then remove the present one if value of present one is greater than the top one else do the opposite.

Code:

class Solution {

public int minCost(String s, int[] neededTime) {

int sum=0;

Stack<Integer> stck = new Stack<>();

stck.push(0);

for(int i=1; i<neededTime.length; i++){

int top = stck.pop();

if(s.charAt(top)==s.charAt(i)){

if(neededTime[top]>neededTime[i]){

stck.push(top);

sum+=neededTime[i];

}else{

stck.push(i);

sum+=neededTime[top];

}

}else{

stck.push(top);

stck.push(i);

}

}

return sum;

}

}

Github Link :<https://lnkd.in/ecwtJeaz>