Lec-92

24/04/24 2:26 PM

312. Burst Balloons

Hard 8826 243 Add to List Share

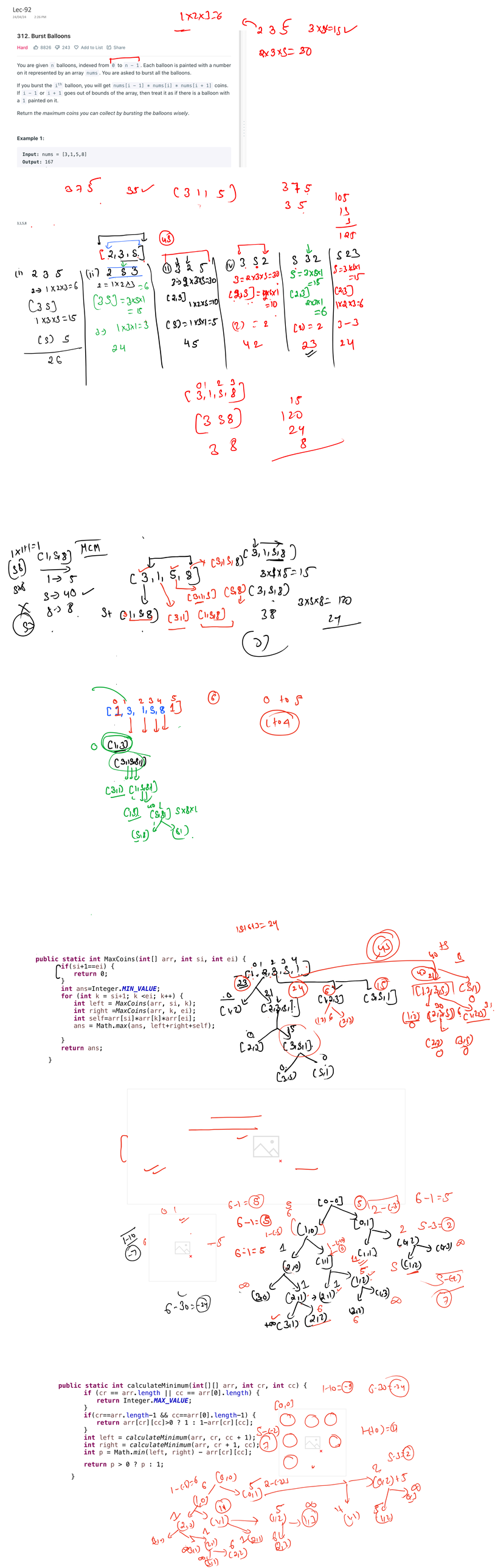
You are given `n` balloons, indexed from `0` to `n - 1`. Each balloon is painted with a number on it represented by an array `nums`. You are asked to burst all the balloons.

If you burst the `i`th balloon, you will get `nums[i - 1] * nums[i] * nums[i + 1]` coins. If `i - 1` or `i + 1` goes out of bounds of the array, then treat it as if there is a balloon with a `1` painted on it.

Return *the maximum coins you can collect by bursting the balloons wisely*.

Example 1:

```
Input: nums = [3,1,5,8]
Output: 167
```



```
public static int MaxCoins(int[] arr, int si, int ei) {
    if(si+1==ei) {
        return 0;
    }
    int ans=Integer.MIN_VALUE;
    for (int k = si+1; k <ei; k++) {
        int left = MaxCoins(arr, si, k);
        int right =MaxCoins(arr, k, ei);
        int self=arr[si]*arr[k]*arr[ei];
        ans = Math.max(ans, left+right+self);

    }
    return ans;
}
```

```
public static int calculateMinimum(int[][] arr, int cr, int cc) {
    if (cr == arr.length || cc == arr[0].length) {
        return Integer.MAX_VALUE;
    }
    if(cr==arr.length-1 && cc==arr[0].length-1) {
        return arr[cr][cc]>0 ? 1 : 1-arr[cr][cc];
    }
    int left = calculateMinimum(arr, cr, cc + 1);
    int right = calculateMinimum(arr, cr + 1, cc);
    int p = Math.min(left, right) - arr[cr][cc];

    return p > 0 ? p : 1;

}
```