Lab10 Solution

YAO ZHAO

Lab10.A: Banning Professional Players

- AUPC (Aim University Persistence Contest) is a famous amateur shooting tournament.
- The match rules are as follow. Each player should make two rounds of shooting. The player can determine the times he or she will fire in the two rounds. However, the player should fire at least once in each round, and the total number of firing should be N.
- Each time the player fires, a score will be produced based on the performance of the shooting. The score of a round is defined as the sum of all scores of the shootings in this round.
- It is stipulated that a player has valid result if and only if the score of each round is not smaller than L. Meanwhile, in order to ban professional players from taking part in the tournament, the score of each round should not be bigger than R, otherwise, the player's result will be canceled.
- **TS1** is a professional shooter who wants to sneak into AUPC. His skill is so perfect and controllable that he knows exactly the **multiset** S of scores of the next N shootings, i.e. $S = \{a_1, a_2, \dots, a_N\}$. He wants to properly distribute the N shootings into two rounds so that the result will be valid and not being cancelled.
- For example, if $S = \{1, 1, 2, 3, 4\}, L = 5, R = 6$, the distribution can be $\{1, 2, 3\}, \{1, 4\}$.
- He wish that you can help him calculate the number of distribution satisfying the previous conditions, modulo $10^9 + 7$. Two distributions are different, if and only if the multisets of scores of the first round are different, or the multisets of scores of the second round are different.

Sample 1 Input

4
3 7
2 3 1 4

$$S = \{2, 3, 1,4\}, L = 3, R = 7$$

$\{4\}\{1,2,3\}$	$\{1,2,3\}\{4\}$
$\{4,1\}\{2,3\}$	$\{2,3\}\{4,1\}$
$\{4,2\}\{1,3\}$	$\{1,3\}\{4,2\}$
$\{4,3\}\{2,1\}$	$\{2,1\}\{4,3\}$
$\{4,1,2\}\{3\}$	$\{3\}\{4,1,2\}$

Sample 1 Output

10

$S = \{a_1, a_2, \dots, a_N\}$
let $Sum = \sum_{i=1}^{N} a_i$

let x_1 is the scores of the 1st round， $x_2$is the scores of the 2nd round.
$x_1 + x_2 = Sum$ → $x_2 = Sum - x_1$ (1)
$L \le x_1 \le R$ (2)
$L \le x_2 \le R$ (3)
(1)(3) → $L \le Sum - x_1 \le R \rightarrow Sum - R \le x_1 \le Sum - L$ (4)
(2)(4) $\max(Sum - R, L) \le x_1 \le \min(Sum - L, R)$

If the multisets of scores of the 1st round is determined, 2nd is determined, so we only think about the number of combinations that satisfy the constraint on x_1 .

The question becomes →
How to pick $a_1 \sim a_N$, so that the sum of the picked items satisfies $\max(Sum - R, L) \le x_1 \le \min(Sum - L, R)$

Since duplicate scores are possible(means ai may equal aj, i≠j), this is a **Bounded Knapsack Problem**(多重背包问题).

Sample Input
6
3 7
1 3 3 1 1 4

$$Sum = 13\ L = 3\ R = 7$$
$$\max(6,3) = 6 \le x_1 \le \min(10,7) = 7$$

$$OPT(i,w) = \begin{cases} 0, & if\ i=0 \\ OPT(i\text{-}1,w) & if\ w_i > \mathrm{w} \\ sum(OPT(i\text{-}1,w), OPT(i\text{-}1, w\text{-}k * w_i)) & 0<w_i<w,\ 0<k\le cnt_i,\ w\text{-}k * w_i \ge 0 \end{cases}$$

w_i

item	cnt
1	3
2	0
3	2
4	1

w / item	0	1	2	3	4	5	6	7
0	1	0	0	0	0	0	0	0
1	1	1	1	1	0	0	0	0
2	1	1	1	1	0	0	0	0
3	1	1	1	2	1	1	2	1
4	1	1	1	2	2	2	3	3

answer = $\sum_{i=\max(Sum-R,L)}^{\min(Sum-L,R)} OPT(i)$

Lab10.B: Rather Be

- Mr. Sorry has a list of N songs. Initially, for each song, Mr. Sorry would grant it a color a_i .
- Mr. Sorry will pick an arbitrary song S as his favourite song. Then, he can perform several operations.
- In each operation, Mr. Sorry would choose some l, r where

  $1 \leq l \leq S \leq r \leq N$

  $a[l] = a[l+1] = a[l+2] = \cdots a[r-1] = a[r]$
- and change $a[l], a[l+1], a[l+2], \ldots, a[r-1], a[r]$ into some other color c.
- Now Mr. Sorry wants to know the minimum operations needed for the entire list to be changed into a single color.

Sample Input 1
4
5 2 2 1

let 1 is his favourite song

5 2 2 1 → 5 2 2 2 → 5 5 5 5

let 2 is his favourite song

5 2 2 1 → 5 5 5 1 → 1 1 1 1

↘ 5 1 1 1 → 5 5 5 5

let 5 is his favourite song

5 2 2 1 → 2 2 2 1 → 1 1 1 1

minimum operations:2

Sample Output 1

2

Sample Input 2

8

4 5 2 2 1 3 5 5

let 2 is his favourite song

4 5 2 2 1 3 5 5 → 4 5 2 2 2 3 5 5 → 4 5 3 3 3 3 5 5

4 5 5 5 5 5 5 5

4 4 4 4 4 4 4 4

Sample Output 2

4

Only one element remains in a continuous the same color interval

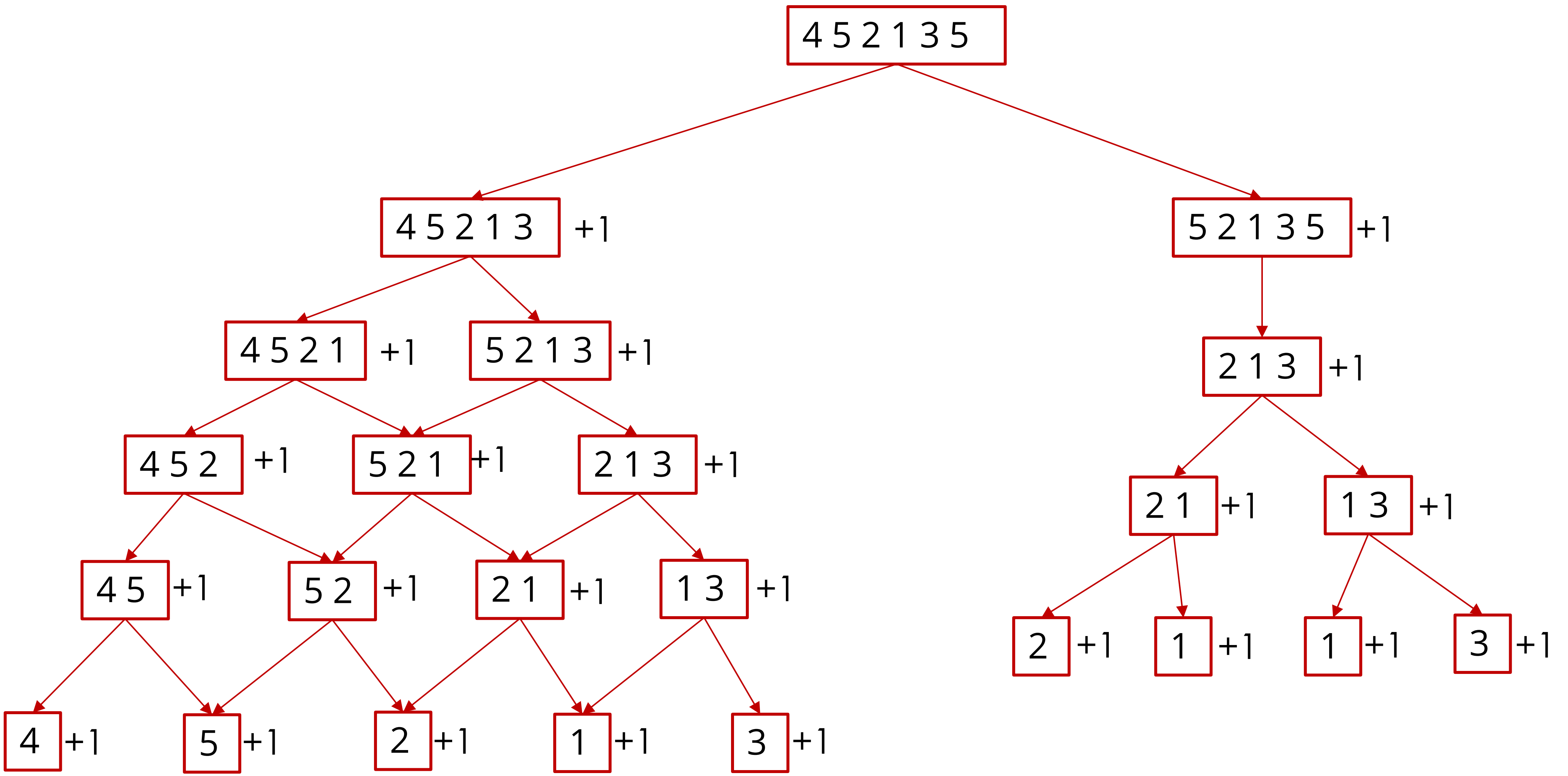
Sample Input 1
4
5 2 2 1

3
5 2 1

Sample Input 2
8
4 5 2 2 1 3 5 5

6
4 5 2 1 3 5

1. Characterize the structure of an optimal solution.



2. Recursively define the value of an optimal solution.

Let opt[i][j] is the minimum total cost of changing a[i]a[i+1]…a[j] to the same color，then：

$$
OPT[i][j] = \begin{cases} \text{if } i=j \quad OPT[i][i]=0 \\ \text{if } a[i]=a[j] \quad OPT[i+1][j-1]+1 \\ \text{if } a[i]\neq a[j] \quad \min\{OPT[i+1][j]+1,\ OPT[i][j-1]+1\} \end{cases}
$$

3. Compute the value of an optimal solution, typically in a bottom-up fashion.

1	2	3	4	5	6
4	5	2	1	3	5

	1	2	3	4	5	6
1	0	1	2	3	4	4
2		0	1	2	3	3
3			0	1	2	3
4				0	1	2
5					0	1
6						0