Zulu and Games - Hackerearth

Tuesday, May 19, 2020 7:54 PM

Zulu loves to play games. Once he started playing a very unusual game. The game consists of *N* levels where each level has two parameters Li & Hi which denotes the lowest and highest energy possible for the ith level of the game respectively. Now before the start of the game, Zulu has the opportunity to increase his energy which is by default *0*. There is a trick called Magic Combo through which Zulu can increase his energy. In Magic Combo, Zulu is allowed to choose several levels as long as their energy levels are non-intersecting. Along with that, after he has finished creating Magic Combo, there should not be any levels left which are not intersecting with the current set of chosen levels and are not picked.

Now, the energy gained after creating a valid Magic Combo is the maximum possible energy amongst all the selected energy levels. As Zulu is very clever, he figured out that he can create many Magic Combos. Can you tell the total possible energy which can be achieved by Zulu after creating all possible Magic Combos? As the total energy value can be very high, return total energy modulus 1000000007 instead.

Note that two Magic Combos are different if there is at least one level which is different among them. Energy intersection occurs between two levels if they both contain at least one common energy number.

Input:
First line contain an integer *N* denoting the number of levels.
Second line contains *N* space separated integers denoting the *L* array where Li denotes the lowest energy possible for the ith level.
Last line contains *N* space separated integers denoting the *H* array where Hi denotes the highest energy possible for the ith level.

Output:
Output in a single line the maximum energy possible modulus 1000000007.

Constraints:
1≤N≤5000
1≤Li,Hi≤109
Li≤Hi

SAMPLE INPUT

```
3
1 3 10
6 8 12
```

SAMPLE OUTPUT

```
24
```

Explanation

Zulu can make *2* magic combos by using combination of level 1,3 and level 2,3. Since, both combos give 12 as the maximum possible energy value, hence answer is 24.

The Code:

```
#include <stdio.h>
#define MOD 1000000007
int *lCopy, *rCopy;

int merge(int l, int r, int* left, int* right) {
int l1, r1, mid, index, i;
mid = (l + r) / 2;
l1 = l;
r1 = mid+1;
index = 0;
while(l1 <= mid && r1 <= r) {
if(right[l1] < right[r1]) {
rCopy[index++] = right[l1++];
lCopy[index-1] = left[l1-1];
} else {
rCopy[index++] = right[r1++];
lCopy[index-1] = left[r1-1];
}
}
while(l1 <= mid) {
rCopy[index++] = right[l1++];
lCopy[index-1] = left[l1-1];
}
while(r1 <= r) {
rCopy[index++] = right[r1++];
lCopy[index-1] = left[r1-1];
}
index = 0;
for(i=l; i<=r; i++) {
left[i] = lCopy[index++];
right[i] = rCopy[index-1];
}
}
int mergeSort(int l, int r, int* left, int* right) {
int mid;
if(l < r) {
mid = (l + r) / 2;
mergeSort(l, mid, left, right);
mergeSort(mid+1, r, left, right);
merge(l, r, left, right);
}
}

int main() {
int N, maxLeft, valid, i, j;
long long int result = 0;
scanf("%d", &N);
int* left = (int*) malloc((N+1) * sizeof(int));
int* right = (int*) malloc((N+1) * sizeof(int));
int* segLen = (int*) calloc((N+1), sizeof(int));
lCopy = (int*) malloc((N+1) * sizeof(int));
rCopy = (int*) malloc((N+1) * sizeof(int));
for(i=1; i<=N; i++) {
scanf("%d", &left[i]);
}
for(i=1; i<=N; i++) {
scanf("%d", &right[i]);
}
mergeSort(1, N, left, right);
left[0] = right[0] = 0;
segLen[0] = 1;
for(i=1; i<=N; i++) {
maxLeft = -1;
for(j=i-1; j>=0; j--) {
if(right[j] >= left[i])
continue;
if(right[j] < maxLeft)
break;
segLen[i] = (segLen[i] + segLen[j]) % MOD;
maxLeft = maxLeft < left[j] ? left[j] : maxLeft;
}
}
result = 0;
for(i=1; i<=N; i++) {
valid = 1;
for(j=i+1; j<=N; j++) {
if(left[j] > right[i]) {
valid = 0;
break;
}
}
if(valid) {
result += (long long int)segLen[i]*right[i] % MOD;
result %= MOD;
}
}
printf("%lld\n", result);
}
```

The Stats:

Score
0.0

Time (sec)
10.17331

Memory (KiB)
64

Language
C