

## 891. Sum of Subsequence Widths

Hard

👍 518

💬 149

❤ Add to List

🔗 Share

The **width** of a sequence is the difference between the maximum and minimum elements in the sequence.

Given an array of integers `nums`, return *the sum of the **widths** of all the non-empty **subsequences** of `nums`*. Since the answer may be very large, return it **modulo**  $10^9 + 7$ .

A **subsequence** is a sequence that can be derived from an array by deleting some or no elements without changing the order of the remaining elements. For example, `[3,6,2,7]` is a subsequence of the array `[0,3,1,6,2,2,7]`.

### Example 1:

**Input:** `nums = [2,1,3]`

**Output:** 6

Explanation: The subsequences are `[1]`, `[2]`, `[3]`, `[2,1]`, `[2,3]`, `[1,3]`, `[2,1,3]`.

The corresponding widths are 0, 0, 0, 1, 1, 2, 2.

The sum of these widths is 6.

### Example 2:

**Input:** `nums = [2]`

**Output:** 0

### Constraints:

- $1 \leq \text{nums.length} \leq 10^5$
- $1 \leq \text{nums}[i] \leq 10^5$

```

#define mod 1000000007

class Solution {
public:

    bool checkBit(int i, int j) {
        return i >> j & 1;
    }

    int sumSubseqWidths(vector<int>& A) {
        int N = A.size(), p = 1<<N;
        vector<int> temp[p];
        for(int i=1; i<p; i++) {
            for(int j=0; j<N; j++) {
                if(checkBit(i, j)) {
                    temp[i].push_back(A[j]);
                }
            }
        }

        int sum = 0;
        for(int i=1; i<p; i++) {
            int mn = INT_MAX, mx = INT_MIN, diff = 0;
            for(int j=0; j<temp[i].size(); j++) {
                mn = min(mn, temp[i][j]);
                mx = max(mx, temp[i][j]);
            }
            diff = mx - mn;
            sum += diff%mod;
        }
        return sum%mod;
    }
};

```

**GOT TLE**

```
1 class Solution {  
2 public:  
3  
4 int sumSubseqWidths(vector<int>& A) {  
5     int N = A.size();  
6     long res = 0, mod = 1000000007;  
7  
8     sort(A.begin(), A.end());  
9     for(int i=0; i<N; i++) {  
10         res = (res + (A[i]*(1<<i)%mod) % mod - (A[i]*(1<<(N-i-1)%mod)) % mod )%mod;  
11     }  
12     return res%mod;  
13 }  
14 };
```

```
1 class Solution {
2 public:
3
4     int sumSubseqWidths(vector<int>& A) {
5         int N = A.size();
6         long mn = 0, mx = 0, mod = 1000000007;
7
8         sort(A.begin(), A.end());
9         for(int i=0; i<N; i++) {
10             mn = (mn*2)%mod;
11             mn = (mn + A[i])%mod;
12         }
13
14         for(int i=N-1; i>=0; i--) {
15             mx = (mx*2)%mod;
16             mx = (mx + A[i])%mod;
17         }
18         return (mx-mn + mod)%mod;
19     }
20 };
```

# #100daysofDSA

---



/rvislive

**Rakesh Vishwakarma**