

2926. Maximum Balanced Subsequence Sum

[My Submissions \(/contest/weekly-contest-370/problems/maximum-balanced-subsequence-sum/submissions/\)](/contest/weekly-contest-370/problems/maximum-balanced-subsequence-sum/submissions/)[Back to Contest \(/contest/weekly-contest-370/\)](/contest/weekly-contest-370/)

You are given a **0-indexed** integer array `nums`.

A **subsequence** of `nums` having length `k` and consisting of **indices** $i_0 < i_1 < \dots < i_{k-1}$ is **balanced** if the following holds:

- $nums[i_j] - nums[i_{j-1}] \geq i_j - i_{j-1}$, for every `j` in the range `[1, k - 1]`.

A **subsequence** of `nums` having length `1` is considered balanced.

Return an integer denoting the **maximum** possible **sum of elements** in a **balanced** subsequence of `nums`.

A **subsequence** of an array is a new **non-empty** array that is formed from the original array by deleting some (possibly none) of the elements without disturbing the relative positions of the remaining elements.

User Accepted:	396
User Tried:	2482
Total Accepted:	452
Total Submissions:	7174
Difficulty:	Hard

Example 1:

Input: `nums = [3,3,5,6]`

Output: 14

Explanation: In this example, the subsequence `[3,5,6]` consisting of indices `0`, `2`, and `3` can be selected.

`nums[2] - nums[0] >= 2 - 0`.

`nums[3] - nums[2] >= 3 - 2`.

Hence, it is a balanced subsequence, and its sum is the maximum among the balanced subsequences of `nums`.

The subsequence consisting of indices `1`, `2`, and `3` is also valid.

It can be shown that it is not possible to get a balanced subsequence with a sum greater than 14.

Example 2:

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

Output: 13

Explanation: In this example, the subsequence `[5,8]` consisting of indices `0` and `3` can be selected.

`nums[3] - nums[0] >= 3 - 0`.

Hence, it is a balanced subsequence, and its sum is the maximum among the balanced subsequences of `nums`.

It can be shown that it is not possible to get a balanced subsequence with a sum greater than 13.

Example 3:

Input: `nums = [-2,-1]`

Output: -1

Explanation: In this example, the subsequence `[-1]` can be selected.

It is a balanced subsequence, and its sum is the maximum among the balanced subsequences of `nums`.

Constraints:

- $1 \leq nums.length \leq 10^5$
- $-10^9 \leq nums[i] \leq 10^9$

[Discuss \(https://leetcode.com/problems/maximum-balanced-subsequence-sum/discuss/\)](https://leetcode.com/problems/maximum-balanced-subsequence-sum/discuss/)

JavaScript



```
1 function Bisect() {  
2   return { insort_right, insort_left, bisect_left, bisect_right }  
}
```

```

3 function insert_right(a, x, lo = 0, hi = null) {
4     lo = bisect_right(a, x, lo, hi);
5     a.splice(lo, 0, x);
6 }
7 function bisect_right(a, x, lo = 0, hi = null) { // > upper_bound
8     if (lo < 0) throw new Error('lo must be non-negative');
9     if (hi == null) hi = a.length;
10    while (lo < hi) {
11        let mid = parseInt((lo + hi) / 2);
12        a[mid] > x ? hi = mid : lo = mid + 1;
13    }
14    return lo;
15 }
16 function insert_left(a, x, lo = 0, hi = null) {
17     lo = bisect_left(a, x, lo, hi);
18     a.splice(lo, 0, x);
19 }
20 function bisect_left(a, x, lo = 0, hi = null) { // >= lower_bound
21     if (lo < 0) throw new Error('lo must be non-negative');
22     if (hi == null) hi = a.length;
23    while (lo < hi) {
24        let mid = parseInt((lo + hi) / 2);
25        a[mid] < x ? lo = mid + 1 : hi = mid;
26    }
27    return lo;
28 }
29 }
30
31 function SegmentTreeRMQArray(A) { // max
32     let n = A.length, h = Math.ceil(Math.log2(n)), len = 2 * 2 ** h, a =
    Array(len).fill(Number.MIN_SAFE_INTEGER);
33     h = 2 ** h;
34     initializeFromArray();
35     return { update, maxx, tree }
36     function initializeFromArray() {
37         for (let i = 0; i < n; i++) a[h + i] = A[i];
38         for (let i = h - 1; i >= 1; i--) pushup(i);
39     }
40     function update(pos, v) {
41         a[h + pos] = v;
42         for (let i = parent(h + pos); i >= 1; i = parent(i)) pushup(i);
43     }
44     function pushup(i) {
45         a[i] = Math.max(a[left(i)], a[right(i)]);
46     }
47     function maxx(l, r) {
48         let max = Number.MIN_SAFE_INTEGER;
49         if (l >= r) return max;
50         l += h;
51         r += h;
52         for (; l < r; l = parent(l), r = parent(r)) {
53             if (l & 1) max = Math.max(max, a[l++]);
54             if (r & 1) max = Math.max(max, a[--r]);
55         }
56         return max;
57     }
58     function parent(i) {
59         return i >> 1;
60     }
61     function left(i) {
62         return 2 * i;
63     }
64     function right(i) {
65         return 2 * i + 1;
66     }
67     function tree() {
68         return a;
69     }
70 }
71


```

```
72 ▼ const maxBalancedSubsequenceSum = (a) => {  
73     let vals = a.map((x, i) => x - i).sort((x, y) => x - y);  
74     vals = [...new Set(vals)];  
75     let n = a.length, st = new SegmentTreeRMQArray(Array(n + 1).fill(Number.MIN_SAFE_INTEGER)), bi = new  
    Bisect();  
76 ▼     for (let i = 0; i < n; i++) {  
77         let v = a[i] - i, idx = bi.bisect_left(vals, v);  
78         let max = st.maxx(0, idx + 1);  
79         if (max < 0) max = 0;  
80         st.update(idx, a[i] + max);  
81     }  
82     return st.maxx(0, n + 1);  
83 };
```

☐ Custom Testcase[Use Example Testcases](#)[Run](#)[Submit](#)**Submission Result: Accepted** (/submissions/detail/1097157473/) ?[More Details > \(/submissions/detail/1097157473/\)](#)

Share your acceptance!

Copyright © 2023 LeetCode

[Help Center \(/support\)](#) | [Jobs \(/jobs\)](#) | [Bug Bounty \(/bugbounty\)](#) | [Online Interview \(/interview/\)](#) | [Students \(/student\)](#) | [Terms \(/terms\)](#) | [Privacy Policy \(/privacy\)](#) [United States \(/region\)](#)

Could not connect to the reCAPTCHA service. Please check your internet connection and reload to get a reCAPTCHA challenge.