

자료구조 2 (민승)

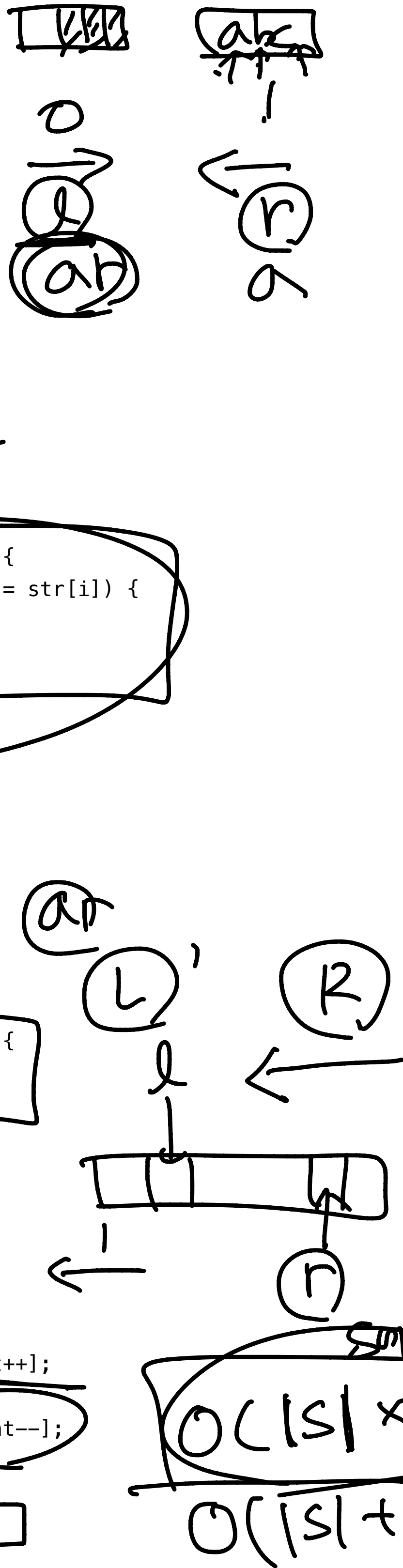
이 코드

최백준 [choi@startlink.io](mailto:choi@startlink.io)

---

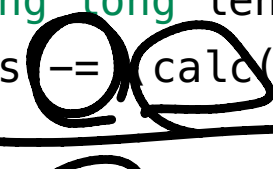
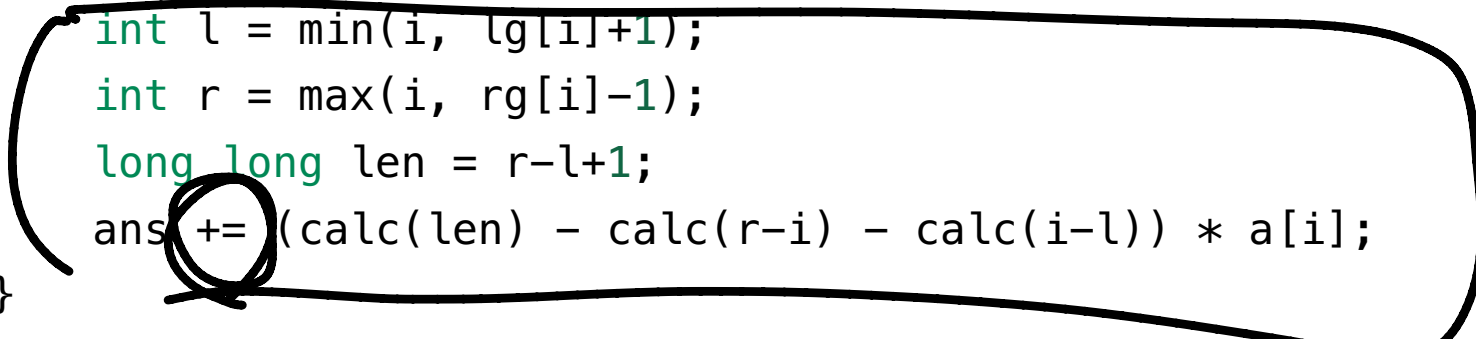
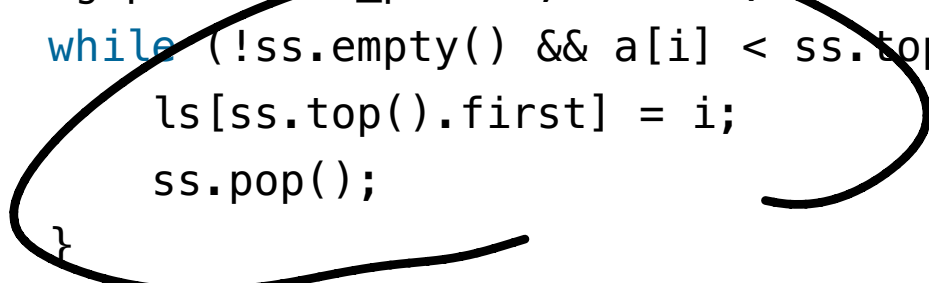
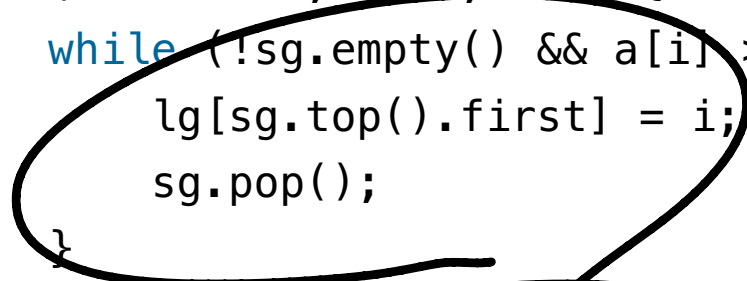
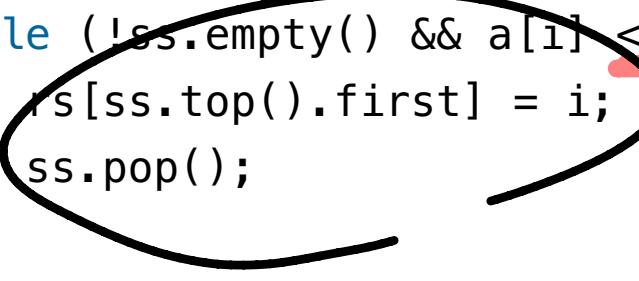
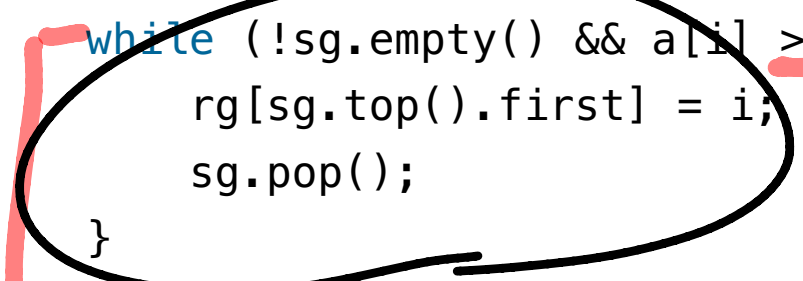
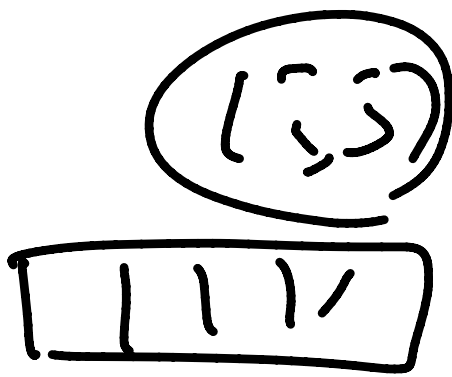
C++14

```
1 #include <cstdio>
2 #include <cstring>
3 using namespace std;
4 char l[300001];
5 char r[300001];
6 int ln=0;
7 int rn=0;
8 char a[31];
9 char ar[31];
10 char b[300001];
11 int n, m;
12 bool check(int where) {
13     char *stack = l;
14     int len = ln;
15     char *str = ar;
16     if (where == 1) {
17         stack = r;
18         len = rn;
19         str = a;
20     }
21     if (len-n < 0) {
22         return false;
23     }
24     for (int i=0; i<n; i++) {
25         if (stack[len-i-1] != str[i]) {
26             return false;
27         }
28     }
29     if (where == 0) {
30         ln -= n;
31     } else {
32         rn -= n;
33     }
34     return true;
35 }
36 int main() {
37     scanf("%s",a);
38     n = strlen(a);
39     for (int i=0; i<n; i++) {
40         ar[i] = a[n-i-1];
41     }
42     scanf("%s",b);
43     m = strlen(b);
44     int left = 0;
45     int right = m-1;
46     int where = 0;
47     while (left <= right) {
48         if (where == 0) {
49             l[ln++] = b[left++];
50         } else {
51             r[rn++] = b[right--];
52         }
53         if (check(where)) {
54             where = 1-where;
55         }
56     }
57     for (int i=ln-1; i>=0; i--) {
58         r[rn++] = l[i];
59     }
60     check(1);
61     for (int i=rn-1; i>=0; i--){
62         putchar(r[i]);
63     }
64     printf("\n");
65     return 0;
66 }
```



C++14

```
1 #include <iostream>
2 #include <vector>
3 #include <algorithm>
4 #include <stack>
5 using namespace std;
6 long long calc(long long n) {
7     return n*(n+1)/2;
8 }
9 int main() {
10     ios_base::sync_with_stdio(false);
11     cin.tie(nullptr);
12     int n;
13     cin >> n;
14     vector<int> a(n+1);
15     vector<int> lg(n+1, 0); // left greater
16     vector<int> rg(n+1, n+1); // right greater
17     vector<int> ls(n+1, 0); // left smaller
18     vector<int> rs(n+1, n+1); // right smaller
19     for (int i=1; i<=n; i++) {
20         cin >> a[i];
21     }
22     {
23         stack<pair<int,int>> sg, ss; // stack greater, smaller
24         sg.push(make_pair(1, a[1]));
25         ss.push(make_pair(1, a[1]));
26         for (int i=2; i<=n; i++) {
27             while (!sg.empty() && a[i] >= sg.top().second) {
28                 rg[sg.top().first] = i;
29                 sg.pop();
30             }
31             sg.push(make_pair(i, a[i]));
32             while (!ss.empty() && a[i] <= ss.top().second) {
33                 ls[ss.top().first] = i;
34                 ss.pop();
35             }
36             ss.push(make_pair(i, a[i]));
37         }
38     }
39     {
40         stack<pair<int,int>> sg, ss; // stack greater, smaller
41         sg.push(make_pair(n, a[n]));
42         ss.push(make_pair(n, a[n]));
43         for (int i=n-1; i>=1; i--) {
44             while (!sg.empty() && a[i] > sg.top().second) {
45                 lg[sg.top().first] = i;
46                 sg.pop();
47             }
48             sg.push(make_pair(i, a[i]));
49             while (!ss.empty() && a[i] < ss.top().second) {
50                 ls[ss.top().first] = i;
51                 ss.pop();
52             }
53             ss.push(make_pair(i, a[i]));
54         }
55     }
56     long long ans = 0;
57     for (int i=1; i<=n; i++) {
58         int l = min(i, lg[i]+1);
59         int r = max(i, rg[i]-1);
60         long long len = r-l+1;
61         ans += (calc(len) - calc(r-i) - calc(i-l)) * a[i];
62     }
63     for (int i=1; i<=n; i++) {
64         int l = min(i, ls[i]+1);
65         int r = max(i, rs[i]-1);
66         long long len = r-l+1;
67         ans -= (calc(len) - calc(r-i) - calc(i-l)) * a[i];
68     }
69     cout << ans << '\n';
70     return 0;
71 }
72
```



C++14

```
1 #include <iostream>
2 #include <vector>
3 #include <cstring>
4 #include <algorithm>
5 #define rank _rank
6 #define size _size
7 using namespace std;
8 int parent[1000001];
9 int rank[1000001];
10 int size[1000001];
11 int Find(int x) {
12     if (x == -1) return -1;
13     } else if (x == parent[x]) {
14         return x;
15     } else {
16         return parent[x] = Find(parent[x]);
17     }
18 }
19
20 long long Union(int x, int y) {
21     x = Find(x);
22     y = Find(y);
23     if (x == y || x == -1 || y == -1) return 0;
24     if (rank[x] < rank[y]) swap(x, y);
25     parent[y] = x;
26     if (rank[x] == rank[y]) {
27         rank[x] = rank[y] + 1;
28     }
29     long long cnt = 1LL * size[x] * size[y];
30     size[x] += size[y];
31     return cnt;
32 }
33
34 void init() {
35     memset(parent, -1, sizeof(parent));
36     memset(size, 0, sizeof(size));
37     memset(rank, 0, sizeof(rank));
38 }
39
40 int main() {
41     ios_base::sync_with_stdio(false);
42     cin.tie(nullptr);
43     int n;
44     cin >> n;
45     vector<pair<int, int>> a(n);
46     for (int i=0; i<n; i++) {
47         cin >> a[i].first;
48         a[i].second = i;
49     }
50     sort(a.begin(), a.end());
51     long long ans = 0;
52     init();
53     for (int i=0; i<n; i++) {
54         int num = a[i].first;
55         int index = a[i].second;
56         parent[index] = index;
57         size[index] = 1;
58         if (index+1 < n) {
59             ans += 1LL * num * Union(index, index+1);
60         }
61         if (index-1 >= 0) {
62             ans += 1LL * num * Union(index, index-1);
63         }
64     }
65     init();
66     for (int i=n-1; i>=0; i--) {
67         int num = a[i].first;
68         int index = a[i].second;
69         parent[index] = index;
70         size[index] = 1;
71         if (index+1 < n) {
72             ans += 1LL * num * Union(index, index+1);
73         }
74         if (index-1 >= 0) {
75             ans += 1LL * num * Union(index, index-1);
76         }
77     }
78     cout << ans << '\n';
79 }
```

결과	메모리	시간	코드 길이
맞았습니다!!	21524 KB	352 ms	1796 B

끝

---



# 코드 플러스

<https://code.plus>

- 슬라이드에 포함된 소스 코드를 보려면 "정보 수정 > 백준 온라인 저지 연동"을 통해 연동한 다음, "백준 온라인 저지"에 로그인해야 합니다.
- 강의 내용에 대한 질문은 코드 플러스의 "질문 게시판"에서 할 수 있습니다.
- 문제와 소스 코드는 슬라이드에 첨부된 링크를 통해서 볼 수 있으며, "백준 온라인 저지"에서 서비스됩니다.
- 슬라이드와 동영상 강의는 코드 플러스 사이트를 통해서만 볼 수 있으며, 동영상 강의의 녹화와 다운로드, 배포와 유통은 저작권법에 의해서 금지되어 있습니다.
- 다른 경로로 이 슬라이드나 동영상 강의를 본 경우에는 [codeplus@startlink.io](mailto:codeplus@startlink.io) 로 이메일 보내주세요.
- 강의 내용, 동영상 강의, 슬라이드, 첨부되어 있는 소스 코드의 저작권은 스타트링크와 최백준에게 있습니다.