

vector 的使用

人员

方俊喆、刘祺、牛同泽、柳力玮、高健桓、赵熙羽、牟茗、齐振玮、夏志赫、武敬哲 到课，辛帅辰、夏硕承 线上

作业检查

上周作业链接: <https://www.luogu.com.cn/contest/208002> (D E 2 道题)

武敬哲、夏志赫、赵熙羽 做了 2 道题

秦显森、牟茗、谢亚锴、齐振玮、柳立玮、高健桓 做了 1 道题

其余同学未做

作业

<https://www.luogu.com.cn/contest/211485> (E F 2 道题)

课堂表现

同学们上课做题表现整体都比较好，不过有同学在老师讲题的时候比着老师写，以后要进行改正，做到听明白了之后自己做。

课堂内容

P5019 [NOIP2018 提高组] 铺设道路

1. 从左往右看

2. 现在在第 i 个位置的话

-> 左边传过来了 $a[i-1]$

1. $a[i] \leq a[i-1]$:

1. 无额外

2. 往右传(不需要做)

2. $a[i] > a[i-1]$

1. 有额外 (= $a[i] - a[i-1]$)

2. 往右传(不需要做)

所有额外值加起来

```
#include <bits/stdc++.h>

using namespace std;

typedef long long LL;
const int maxn = 1e5 + 5;
int w[maxn];

int main()
{
    int n; cin >> n;
    for (int i = 1; i <= n; ++i) cin >> w[i];

    LL res = 0;
    for (int i = 1; i <= n; ++i) res += max(0, w[i]-w[i-1]);
    cout << res << endl;
    return 0;
}
```

P1109 学生分组

1. $\text{sum} < l * n$ 或者 $\text{sum} > r * n$ $\rightarrow -1$
2. 否则，考虑一共需要进多少人，一共需要出多少人，输出最大值即可

```
#include <bits/stdc++.h>

using namespace std;

typedef long long LL;
const int maxn = 50 + 5;
int w[maxn];

int main()
{
    int n, sum = 0; cin >> n;
    for (int i = 1; i <= n; ++i) cin >> w[i], sum += w[i];
    int l, r; cin >> l >> r;

    if (sum < l * n || sum > r * n) { cout << -1 << endl; return 0; }

    int res1 = 0, res2 = 0;
    for (int i = 1; i <= n; ++i) {
        res1 += max(0, l - w[i]), res2 += max(0, w[i] - r);
    }
    cout << max(res1, res2) << endl;
}
```

```
    return 0;
}
```

P1324 矩形分割

把横竖的值存到结构体里, 然后把结构体内部按照从大到小排序, 最后从前到后扫一遍即可

```
#include <bits/stdc++.h>

using namespace std;

typedef long long LL;
const int maxn = 4000 + 5;
struct node {
    int value, op;
} w[maxn];

bool cmp(node p, node q) {
    return p.value > q.value;
}

int main()
{
    int n, m; cin >> n >> m;
    int id = 1;
    for (int i = 1; i <= n-1; i++) {
        int x;
        cin >> x;
        w[id] = {x, 0};
        id++;
    }
    for (int i = 1; i <= m-1; i++) {
        int x;
        cin >> x;
        w[id] = {x, 1};
        id++;
    }

    id--;
    sort(w+1, w+id+1, cmp);

    LL res = 0;
    int h = 0, s = 0;
    for (int i = 1; i <= id; i++) {
        if (w[i].op == 0) {
            res += (LL)w[i].value * (s+1);
            h++;
        } else {
            res += (LL)w[i].value * (h+1);
            s++;
        }
    }
}
```

```

    }
    cout << res << endl;
    return 0;
}

```

vector

```

#include <vector>

vector<int> vec;
vec.push_back(11);
int n = vec.size();

for (int i = 0; i < (int)vec.size(); i++) {
    cout << vec[i] << " ";
}
cout << endl;

for (int i : vec) {
    cout << i << " ";
}
cout << endl;

vec.back()
vec.pop_back()
vec.empty()

```

U499172 找每个数出现的第 k 个位置

```

#include <bits/stdc++.h>

using namespace std;

const int maxn = 5e5 + 5;
vector<int> vec[maxn];

int main()
{
    int n, q; cin >> n >> q;
    for (int i = 1; i <= n; i++) {
        int x; cin >> x; // 第 i 个位置的值是 x
        // i -> vec[x]
        vec[x].push_back(i);
    }

    while (q -- ) {
        int x, k; cin >> x >> k;
        // vec[x]: x 这个数出现的所有位置
    }
}

```

```
        // vec[x][0], vec[x][1], ... , vec[x][k-1]
        cout << vec[x][k-1] << endl;
    }
    return 0;
}
```