最长上升子序列

人员

陶汇笙、郭栩睿、洪晨栋、洪晨棋、崔宸赫、王恩泽、于霄龙、于家瑞 到课

上周作业检查

上周作业链接: https://cppoj.kids123code.com/contest/2326

#	用户名	姓名	编程分	时间	A	B	C	D	E
1	yuxiaolong	于霄龙	500	297	100	100	100	100	100
2	taohuisheng	陶汇笙	500	301	100	100	100	100	100
3	wangenze	王恩泽	500	3408	100	100	100	100	100
4	hongchendong	洪晨栋	400	174	100	100	100	100	
5	guoxurui	郭栩睿	400	179	100	100	100	100	
6	yujiarui	于家瑞	360	114	100	100	100	60	
7	hongchenqi	洪晨棋	300	122	100	100	100	0	
8	cuichenhe	崔宸赫	300	133	100	100	100		

本周作业

https://cppoj.kids123code.com/contest/2450 (课上讲了 A ~ D 这些题, 课后作业是 E 题)

课堂表现

今天着重讲了 最长上升子序列 的 nlogn 做法, 核心代码就 2 行, 需要同学们背熟。

课堂内容

四方定理 (上周作业)

f[i][j]: 对于 i 这个数, 如果他用 j 个整数的平方和构成时, 一共有多少不同的方案

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

using namespace std;

typedef long long LL;
const int maxn = 32768 + 5;
LL f[maxn][5];

int main()
{
    f[0][0] = 1;
```

```
    for (int i = 1; i*i < maxn; ++i) {
        for (int j = i*i; j < maxn; ++j) {
            for (int k = 1; k <= 4; ++k) f[j][k] += f[j-i*i][k-1];
        }
    }

    int T; cin >> T;
    while (T -- ) {
        int x; cin >> x;
        int res = 0;
        for (int i = 1; i <= 4; ++i) res += f[x][i];
        cout << res << endl;
    }
    return 0;
}
```

信封问题

错排, f[i]: 对 i 封信进行错排时, 共有多少不同的方案

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

using namespace std;

typedef long long LL;
const int maxn = 20 + 5;
LL f[maxn];

int main()
{
  int n; cin >> n;
  f[1] = 0, f[2] = 1;
  for (int i = 3; i <= n; ++i) {
    f[i] = (i-1) * (f[i-2] + f[i-1]);
  }
  cout << f[n] << endl;
  return 0;
}
```

最长上升子序列

用二分优化最长上升子序列, 实现 nlogn 求最长上升子序列

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

using namespace std;

const int maxn = 1e6 + 5;
```

```
int main()
{
  int n; cin >> n;
  vector<int> vec;
  while (n -- ) {
    int x; cin >> x;
    if (vec.empty() || x>vec.back()) vec.push_back(x);
    else *lower_bound(vec.begin(), vec.end(),x) = x;
  }

  cout << vec.size() << endl;
  return 0;
}
```

两个排列的最长公共子序列

把两个排列的 最长公共子序列 问题, 转化为 最长上升子序列 问题

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

using namespace std;

const int maxn = 1e5 + 5;
int a[maxn], b[maxn], c[maxn];

int main()
{
  int n; cin >> n;
  for (int i = 1; i <= n; ++i) cin >> a[i], c[a[i]] = i;
  for (int i = 1; i <= n; ++i) cin >> b[i], b[i] = c[b[i]];

  vector<int> vec;
  for (int i = 1; i <= n; ++i) {
    if (vec.empty() || b[i]>vec.back()) vec.push_back(b[i]);
    else *lower_bound(vec.begin(), vec.end(), b[i]) = b[i];
  }
  cout << vec.size() << endl;
  return 0;
}
```

签到题

维护一个 f[i] 和 s[i] 数组, 其中:

f[i]: 以 i 结尾的最长连续上升子串的长度

s[i]: 以 i 开头的最长连续上升子串的长度

利用 f[i] 和 s[i] 来求后续答案

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

using namespace std;

const int maxn = 1e6 + 5;
int w[maxn];
int f[maxn];  // f[i]: 以 i 结尾的最长连续上升子串的长度
int s[maxn];  // s[i]: 以 i 开头的最长连续上升子串的长度

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

  f[1] = 1;
  for (int i = 2; i <= n; ++i) {
    if (w[i] > w[i-1]) f[i] = f[i-1] + 1;
    else f[i] = 1;
  }

  s[n] = 1;
  for (int i = n-1; i >= 1; --i) {
    if (w[i] < w[i+1]) s[i] = s[i+1] + 1;
    else s[i] = 1;
  }

  int res = 0;
  for (int i = 1; i <= n; ++i) res = max(res, f[i]);

  for (int i = 2; i <= n; ++i) res = max(res, f[i-1]+1);

  for (int i = 1; i <= n-1; ++i) res = max(res, s[i+1]+1);

  for (int i = 2; i <= n-1; ++i) {
    if (w[i+1] - w[i-1] >= 2) res = max(res, f[i-1]+1+s[i+1]);
  }
  cout << res << endl;
  return 0;
}
```