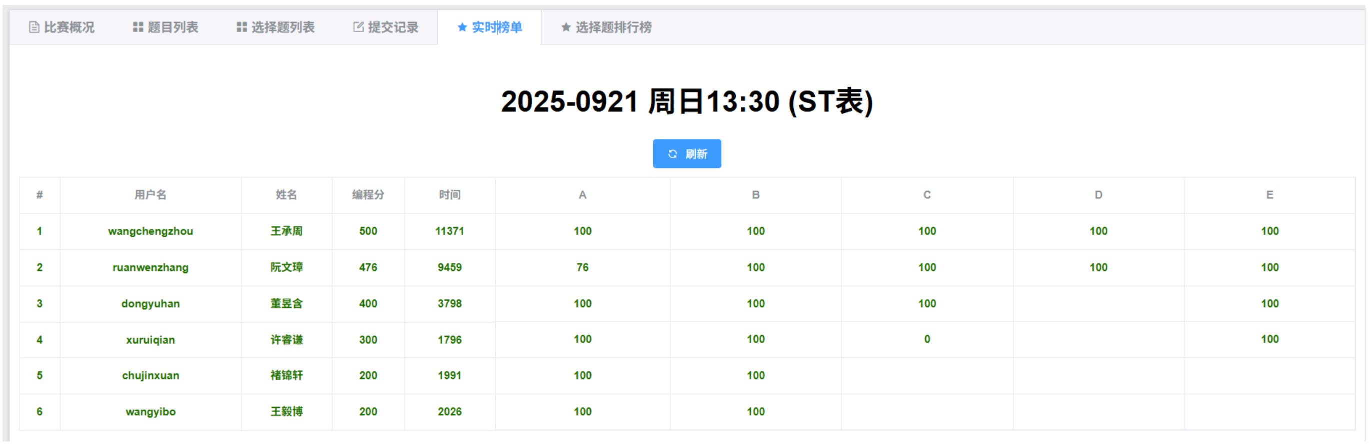
综合练习

人员

王毅博、阮文璋、褚锦轩、王承周、许睿谦 到课

上周作业检查

上周作业链接: https://cppoj.kids123code.com/contest/847



2025-0921 周日13:30 (ST表)

#	用户名	姓名	编程分	时间	A	B	C	D	E
1	wangchengzhou	王承周	500	11371	100	100	100	100	100
2	ruanwenzhang	阮文璋	476	9459	76	100	100	100	100
3	dongyuhan	董昱含	400	3798	100	100	100		100
4	xuruiqian	许睿谦	300	1796	100	100	0		100
5	chujinxuan	褚锦轩	200	1991	100	100			
6	wangyibo	王毅博	200	2026	100	100			

本周作业

https://cppoj.kids123code.com/contest/968 (课上讲了 A ~ C 题, 课后作业是 D 题)

课堂表现

今天的 A 题是一道裸的搜索题, 就是做两遍 搜索 找最远点, 同学们课上做 A 题整体都做的不太好, 课下要好好复习一下搜索的题目, 把之前的搜索题可以再做一做。

其中, 褚锦轩同学这节课做题表现比较好, 提出表扬！

课堂内容

[USACO07JAN] Balanced Lineup G (上周作业)

ST 表模板, 用 区间最大值 减去 区间最小值

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

using namespace std;

const int N = 5e4 + 5, M = 20;
int w[N], f1[N][M], f2[N][M], _lg2[N];

int get(int l, int r, bool flag) {
  int len = r - l + 1;
  int k = _lg2[len];
  if (flag) return max(f1[l][k], f1[r-(1<<k)+1][k]);
  return min(f2[l][k], f2[r-(1<<k)+1][k]);
}

int main()
{
  for (int i = 0; (1<<i) < N; ++i) _lg2[1<<i] = i;
  for (int i = 1; i < N; ++i) {
    if (!_lg2[i]) _lg2[i] = _lg2[i-1];
  }

  int n, m; cin >> n >> m;
  for (int i = 1; i <= n; ++i) cin >> w[i], f1[i][0] = f2[i][0] = w[i];
  for (int k = 1; k < M; ++k) {
    for (int i = 1; i+(1<<k)-1 <= n; ++i) {
      f1[i][k] = max(f1[i][k-1], f1[i+(1<<(k-1))][k-1]);
      f2[i][k] = min(f2[i][k-1], f2[i+(1<<(k-1))][k-1]);
    }
  }

  while (m -- ) {
    int l, r; cin >> l >> r;
    cout << get(l,r,true) - get(l,r,false) << endl;
  }
  return 0;
}
```

[蓝桥杯 2013 省 A] 大臣的旅费

两次搜索, 求树的带权直径

先从任意一点出发, 找到离他最远的点, 从这个点出发再搜一遍

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

using namespace std;

typedef long long LL;
const int maxn = 1e5 + 5;
struct node {
  int to, value;
};
vector<node> vec[maxn];
int dis[maxn];

void bfs(int x) {
  memset(dis, -1, sizeof(dis));

  queue<int> q; q.push(x); dis[x] = 0;
  while (!q.empty()) {
    int u = q.front(); q.pop();
    for (node it : vec[u]) {
      if (dis[it.to] != -1) continue;
      q.push(it.to), dis[it.to] = dis[u]+it.value;
    }
  }
}

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

    bfs(1);

    int id = 1;
    for (int i = 2; i <= n; ++i) {
    if (dis[i] > dis[id]) id = i;
    }

    bfs(id);

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

    cout << res*10 + (LL)res*(res+1)/2 << endl;
  return 0;
}
```

[JRKSJ R1] JFCA

用 ST 表维护区间最大的 a 值

破环成链, 把数组拷贝 3 份, 针对中间这一份数组, 向左或向右进行二分, 找到最近的符合要求的点 j

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

using namespace std;

const int N = 1e5 + 5, M = 20;
const int inf = 0x3f3f3f3f;
int a[3*N], b[3*N], f[3*N][M], _lg2[3*N];

int get_max(int l, int r) {
  int k = _lg2[r-l+1];
  return max(f[l][k], f[r-(1<<k)+1][k]);
}

int main()
```

```
{
  for (int i = 0; (1<<i) < 3*N; ++i) _lg2[1<<i] = i;
  for (int i = 1; i < 3*N; ++i) {
    if (!_lg2[i]) _lg2[i] = _lg2[i-1];
  }

  int n; cin >> n;
  for (int i = 1; i <= n; ++i) cin >> a[i], a[i+2*n] = a[i+n] = a[i];
  for (int i = 1; i <= n; ++i) cin >> b[i], b[i+2*n] = b[i+n] = b[i];

  for (int i = 1; i <= 3*n; ++i) f[i][0] = a[i];
  for (int k = 1; k < M; ++k) {
    for (int i = 1; i+(1<<k)-1 <= 3*n; ++i) {
      f[i][k] = max(f[i][k-1], f[i+(1<<(k-1))][k-1]);
    }
  }

  for (int i = n+1; i <= n+n; ++i) {
    int lLen = inf, rLen = inf;
    if (get_max(i-n+1,i-1) >= b[i]) {
      int l = i-n+1, r = i-1;
      while (l <= r) {
        int mid = (l + r) / 2;
        if (get_max(mid,i-1) >= b[i]) l = mid+1;
        else r = mid-1;
      }
      lLen = i - r;
    }

    if (get_max(i+1, i+n-1) >= b[i]) {
      int l = i+1, r = i+n-1;
      while (l <= r) {
        int mid = (l + r) / 2;
        if (get_max(i+1,mid) >= b[i]) r = mid-1;
        else l = mid+1;
      }
      rLen = l - i;
    }

    int res = min(lLen, rLen);
    cout << (res==inf ? -1 : res) << " ";
  }
  cout << endl;
  return 0;
}
```

NAND repeatedly

线性 dp

f[i][0] 代表以第 i 项结尾时有多少 0, f[i][1] 代表以第 i 项结尾时有多少 1

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

using namespace std;

typedef long long LL;
const int maxn = 1e6 + 5;
char s[maxn];
LL f[maxn][2];

int main()
{
  int n; cin >> n >> (s+1);

  LL res = 0;
  for (int i = 1; i <= n; ++i) {
    if (s[i] == '0') {
      f[i][0] = 1, f[i][1] = f[i-1][0] + f[i-1][1];
    } else {
      f[i][0] = f[i-1][1], f[i][1] = f[i-1][0]+1;
    }
    res += f[i][1];
  }
  cout << res << endl;
  return 0;
}
```