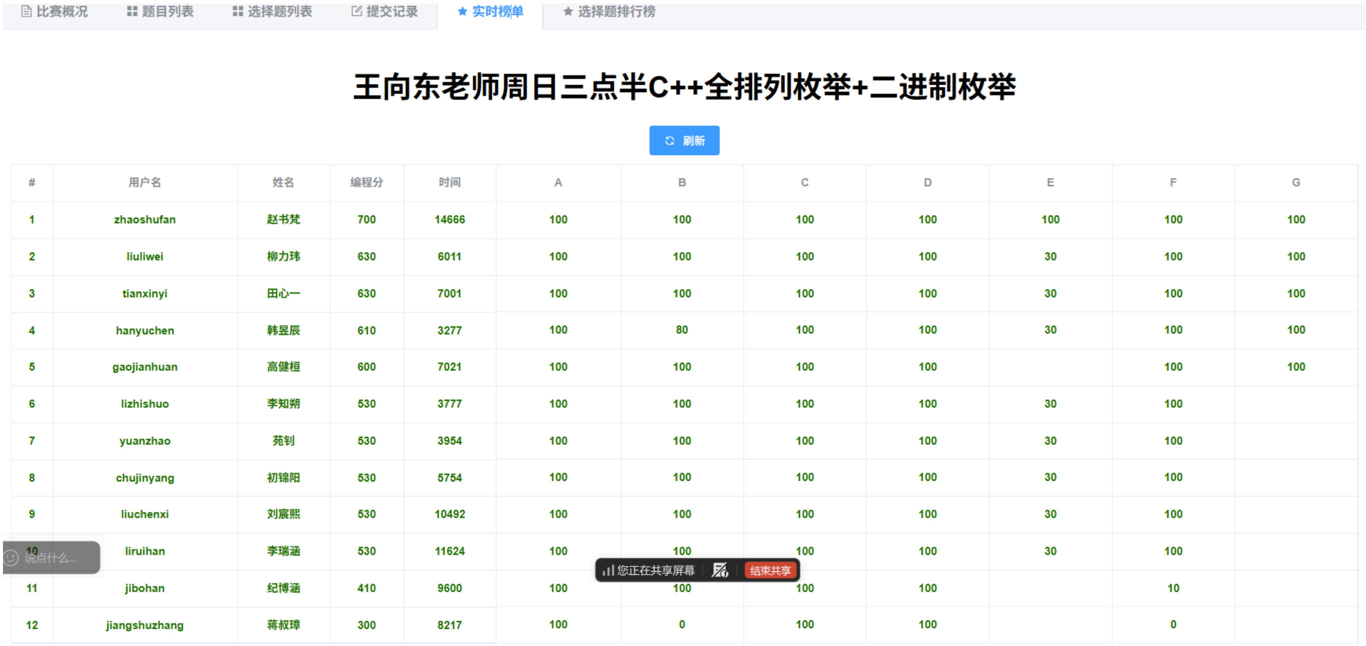
记忆化搜索

人员

柳力玮、田心一、刘宸熙、赵书梵、李瑞涵、李知朔、纪博涵、蒋叔璋、初锦阳、蒋承恩 到课, 苑钊、高健桓线上

上周作业检查

上周作业链接: https://cppoj.kids123code.com/contest/1790



王向东老师周日三点半C++全排列枚举+二进制枚举

#	用户名	姓名	编程分	时间	A	B	C	D	E	F	G
1	zhaoshufan	赵书梵	700	14666	100	100	100	100	100	100	100
2	liuliwei	柳力玮	630	6011	100	100	100	100	30	100	100
3	tianxinyi	田心一	630	7001	100	100	100	100	30	100	100
4	hanyuchen	韩昱辰	610	3277	100	80	100	100	30	100	100
5	gaojianhuan	高健桓	600	7021	100	100	100	100		100	100
6	lizhishuo	李知朔	530	3777	100	100	100	100	30	100	
7	yuanzhao	苑钊	530	3954	100	100	100	100	30	100	
8	chujinyang	初锦阳	530	5754	100	100	100	100	30	100	
9	liuchenxi	刘宸熙	530	10492	100	100	100	100	30	100	
10	liruihan	李瑞涵	530	11624	100	100	100	100	30	100	
11	jibohan	纪博涵	410	9600	100	100	100	100		10	
12	jiangshuzhang	蒋叔璋	300	8217	100	0	100	100		0	

本周作业

https://cppoj.kids123code.com/contest/1912 (课上讲了 A ~ E 题, 课后作业是 F 题)

课堂表现

今天的 D 题比较难一些, 同学们课上普遍做的都不是太好, 课下需要再复习复习这道题。

课堂内容

[蓝桥杯 2023 省 B] 飞机降落 (上周作业)

全排列枚举所有飞机排列的可能性, 看是否存在有一种合法情况

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

using namespace std;

const int maxn = 10 + 5;
int T[maxn], D[maxn], L[maxn];
int w[maxn];
int n;

bool check() {
  int last = -1;
  for (int i = 1; i <= n; ++i) {
    int id = w[i];
    int l = T[id], r = T[id] + D[id];
    if (last > r) return false;
    last = max(last, l) + L[id];
  }
  return true;
}

void solve() {
  cin >> n;
  for (int i = 1; i <= n; ++i) cin >> T[i] >> D[i] >> L[i];

  for (int i = 1; i <= n; ++i) w[i] = i;
  do {
    if (check()) { cout << "YES" << endl; return; }
  } while (next_permutation(w+1, w+n+1));
  cout << "NO" << endl;
}

int main()
{
  int T; cin >> T;
  while (T -- ) solve();
  return 0;
}
```

斐波那契数列

记忆化搜索模板, 把 fib(i) 的值存放在 f[i] 中, 后续就不用每次都去搜了

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

using namespace std;

const int maxn = 30 + 5;
int f[maxn];

int fib(int n) {
  if (n==1 || n==2) return 1;
  if (f[n]) return f[n];
  f[n] = fib(n-1) + fib(n-2);
  return f[n];
}

int main()
```

```
{
  int T; cin >> T;
  while (T -- ) {
    int n; cin >> n;
    cout << fib(n) << endl;
  }
  return 0;
}
```

Function

记忆化搜索模板题, 直接搜即可

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

using namespace std;

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

LL dfs(LL a, LL b, LL c) {
    if (a<=0 || b<=0 || c<=0) return 1;
    if (a>20 || b>20 || c>20) return dfs(20, 20, 20);
    if (st[a][b][c]) return f[a][b][c];
    st[a][b][c] = true;

    if (a<b && b<c) f[a][b][c] = dfs(a,b,c-1)+dfs(a,b-1,c-1)-dfs(a,b-1,c);
    else f[a][b][c] = dfs(a-1,b,c)+dfs(a-1,b-1,c)+dfs(a-1,b,c-1)-dfs(a-1,b-1,c-1);
    return f[a][b][c];
}

int main()
{
    LL a, b, c;
    while (true) {
        cin >> a >> b >> c;
        if (a==-1 && b==-1 && c==-1) break;
        printf("w(%lld, %lld, %lld) = %lld\n", a, b, c, dfs(a,b,c));
    }
    return 0;
}
```

Yet Another Recursive Function

dfs(n): 求 f[n] 对应的值是多少, 并把计算出来的值用 map 存下来

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

using namespace std;

typedef long long LL;
map<LL, LL> mp;

LL dfs(LL n) {
  if (n == 0) return 1;
  if (mp.count(n)) return mp[n];
  mp[n] = dfs(n/2) + dfs(n/3);
  return mp[n];
}

int main()
{
  LL n;
  cin >> n;
  cout << dfs(n) << endl;
  return 0;
}
```

[SHOI2002] 滑雪

dfs(i,j): 从 (i,j) 这个点往后走, 最多能走多少步

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

using namespace std;

const int maxn = 100 + 5;
int w[maxn][maxn], f[maxn][maxn];
int n, m;
int dx[] = {-1, 1, 0, 0}, dy[] = {0, 0, -1, 1};

int dfs(int x, int y) {
  if (f[x][y]) return f[x][y];
  int maxx = 0;
  for (int i = 0; i < 4; ++i) {
    int nx = x+dx[i], ny = y+dy[i];
    if (nx>=1 && nx<=n && ny>=1 && ny<=m && w[x][y]>w[nx][ny]) maxx = max(maxx,
dfs(nx,ny));
  }
  f[x][y] = maxx+1;
  return f[x][y];
}

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

  int res = 0;
  for (int i = 1; i <= n; ++i) {
    for (int j = 1; j <= m; ++j) res = max(res, dfs(i,j));
  }
  cout << res << endl;
  return 0;
}
```

Socks 2

当 k 是偶数时, 直接贪心两个两个选

当 k 是奇数时, 先维护一个两两差的前缀和, 再维护一个两两差的后缀和, 然后枚举删除哪个。

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

using namespace std;

typedef long long LL;
const int maxn = 4e5 + 5;
const LL inf = 0x3f3f3f3f3f3f3f3f;
bool st[maxn];
int w[maxn];
LL pre[maxn], suf[maxn];

int main()
{
    int n, k; cin >> n >> k;
    while (k -- ) { int x; cin >> x; st[x] = true; }

    int id = 0;
    for (int i = 1; i <= n; ++i) {
        ++id; w[id] = i;
        if (!st[i]) { ++id; w[id] = i; }
    }

    if (id%2 == 0) {
        LL res = 0;
        for (int i = 2; i <= id; i += 2) res += w[i]-w[i-1];
        cout << res << endl;
    } else {
        for (int i = 2; i <= id; i += 2) pre[i] = pre[i-2] + w[i]-w[i-1];
        for (int i = id-1; i >= 1; i -= 2) suf[i] = suf[i+2] + w[i+1]-w[i];

        LL res = inf;
        for (int i = 1; i <= id; ++i) {
            if (i&1) res = min(res, pre[i-1]+suf[i+1]);
            else res = min(res, pre[i-2]+suf[i+2]+w[i+1]-w[i-1]);
        }
        cout << res << endl;
    }
    return 0;
}
```