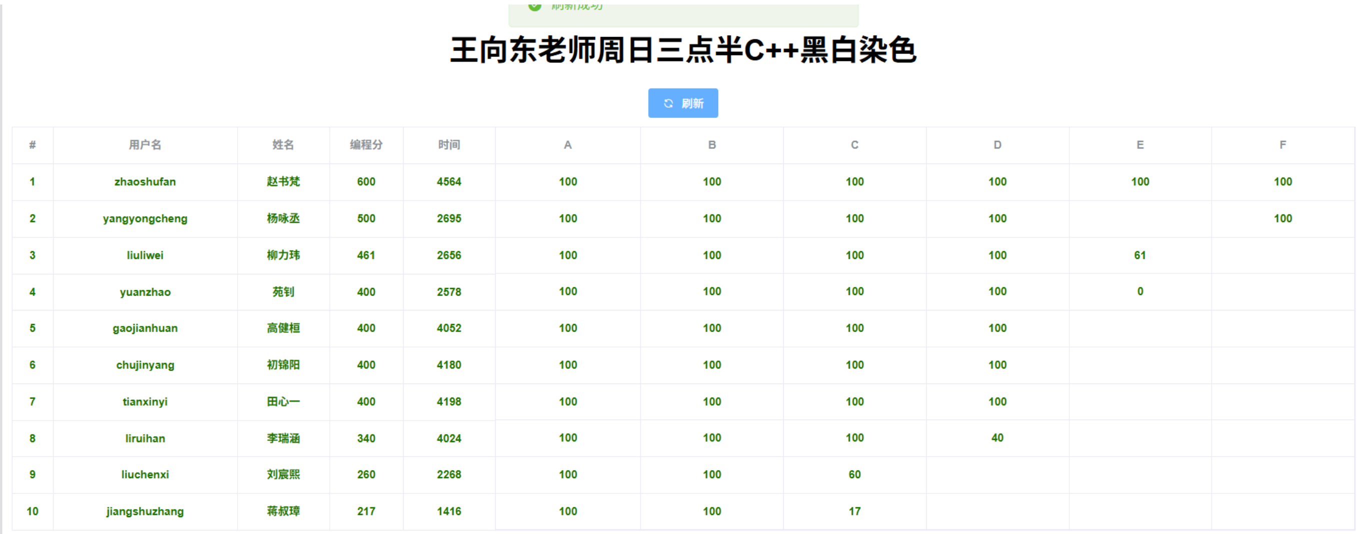
归并排序

人员

初锦阳、纪博涵、柳力玮、李瑞涵 到课, 高健桓、杨咏丞 线上

上周作业检查

上周作业链接: https://cppoj.kids123code.com/contest/2686



刷新成功

王向东老师周日三点半C++黑白染色

#	用户名	姓名	编程分	时间	A	B	C	D	E	F
1	zhaoshufan	赵书梵	600	4564	100	100	100	100	100	100
2	yangyongcheng	杨咏丞	500	2695	100	100	100	100		100
3	liuliwei	柳力玮	461	2656	100	100	100	100	61	
4	yuanzhao	苑钊	400	2578	100	100	100	100	0	
5	gaojianhuan	高健桓	400	4052	100	100	100	100		
6	chujinyang	初锦阳	400	4180	100	100	100	100		
7	tianxinyi	田心一	400	4198	100	100	100	100		
8	liruihan	李瑞涵	340	4024	100	100	100	40		
9	liuchenxi	刘宸熙	260	2268	100	100	60			
10	jiangshuzhang	蒋叔璋	217	1416	100	100	17			

本周作业

https://cppoj.kids123code.com/contest/2903 (课上讲了 A ~ D 题, 课后作业是 D 题必做)

课堂表现

今天课上来的几位同学做题表现都比较好, 希望继续保持。

课堂内容

【模板】排序

归并排序 模板

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

using namespace std;

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

void merge_sort(int l, int r) {
    if (l >= r) return;
    int mid = (l+r) / 2;
    merge_sort(l, mid), merge_sort(mid+1, r);

    int i = l, j = mid+1;
    vector<int> vec;
    while (i<=mid && j<=r) {
        if (a[i] <= a[j]) vec.push_back(a[i]), i++;
        else vec.push_back(a[j]), j++;
    }

    while (i <= mid) vec.push_back(a[i]), i++;
    while (j <= r) vec.push_back(a[j]), j++;

    for (int i = l, j = 0; i <= r; i++, j++) a[i] = vec[j];
}

int main()
{
    int n; cin >> n;
    for (int i = 1; i <= n; i++) cin >> a[i];
    merge_sort(1, n);
    for (int i = 1; i <= n; i++) cout << a[i] << " ";
    cout << endl;
    return 0;
}
```

逆序对

归并排序求逆序对

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

using namespace std;

typedef long long LL;
const int maxn = 5e5 + 5;
int a[maxn];
LL res = 0;

void merge_sort(int l, int r) {
    if (l >= r) return;

    int mid = (l+r) / 2;
    merge_sort(l, mid), merge_sort(mid+1, r);

    int i = l, j = mid+1;
    vector<int> vec;
    while (i<=mid && j<=r) {
        if (a[i] <= a[j]) vec.push_back(a[i]), i++;
        else vec.push_back(a[j]), j++, res += mid-i+1;
    }
    while (i <= mid) vec.push_back(a[i]), i++;
    while (j <= r) vec.push_back(a[j]), j++;

    for (int i = l, j = 0; i <= r; i++, j++) a[i] = vec[j];
}

int main()
{
    int n; cin >> n;
    for (int i = 1; i <= n; i++) cin >> a[i];
    merge_sort(1, n);
    cout << res << endl;
    return 0;
}
```

取数游戏

搜索 + 剪枝

当一个点要选择时, 把周围 8 个位置标记

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

using namespace std;

const int maxn = 6 + 5;
int w[maxn][maxn];
int n, m, res;
int cnt[maxn][maxn];
int dx[] = {-1, -1, -1, 0, 0, 0, 1, 1, 1}, dy[] = {-1, 0, 1, -1, 0, 1, -1, 0, 1};

void dfs(int x, int y, int sum) {
  if (y == m+1) return dfs(x+1, 1, sum);
  if (x == n+1) { res = max(res, sum); return; }

  dfs(x, y+1, sum);

  if (!cnt[x][y]) {
    for (int i = 0; i < 9; ++i) {
      int nx = x+dx[i], ny = y+dy[i];
      if (nx>=1 && nx<=n && ny>=1 && ny<=m) cnt[nx][ny]++;
    }
    dfs(x, y+1, sum+w[x][y]);
    for (int i = 0; i < 9; ++i) {
      int nx = x+dx[i], ny = y+dy[i];
      if (nx>=1 && nx<=n && ny>=1 && ny<=m) cnt[nx][ny]--;
    }
  }
}
```

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

  memset(cnt, 0, sizeof(cnt)); res = 0;
  dfs(1, 1, 0);
  cout << res << endl;
}

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

[USACO1.4] 母亲的牛奶 Mother's Milk

记忆化搜索

标记所有到达过的情况

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

using namespace std;

const int maxn = 20 + 5;
bool f[maxn][maxn][maxn];
vector<int> vec;
int A, B, C;

void dfs(int x, int y, int z) {
  if (f[x][y][z]) return;
  f[x][y][z] = true;
  if (x==0) vec.push_back(z);

  if (x+y <= B) dfs(0, x+y, z);
  else dfs(x-(B-y), B, z);

  if (x+z <= C) dfs(0, y, x+z);
  else dfs(x-(C-z), y, C);

  if (y+x <= A) dfs(y+x, 0, z);
  else dfs(A, y-(A-x), z);

  if (y+z <= C) dfs(x, 0, y+z);
  else dfs(x, y-(C-z), C);

  if (z+x <= A) dfs(z+x, y, 0);
```

```cpp
  else dfs(A, y, z-(A-x));

  if (z+y <= B) dfs(x, z+y, 0);
  else dfs(x, B, z-(B-y));
}

int main()
{
  cin >> A >> B >> C;
  dfs(0, 0, C);
  sort(vec.begin(), vec.end());
  for (int i : vec) cout << i << " ";
  cout << endl;
  return 0;
}
```