综合混练

人员

赵广宇、刘智予、李政毅、张皓宁、韩鸿钜、许岩、方冠霖、金一航、曹承贤、陈瀚霄、黄诗琦 到课

作业

https://vjudge.net/contest/693751 (课上讲了 A、B 题, 课后必做作业是 C 题, 选做作业是 D 题)

课堂表现

同学们课上听讲思路基本都听会了, 但是代码还没有完全写对, 课下要把没写完的题目重新完善一下。

课堂内容

Gym - 102769E

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

using namespace std;

typedef long long LL;
const int maxn = 2e5 + 5;
struct node {
  int value, id;
  bool operator < (const node& p) const { return value < p.value; }
} w[maxn*2];

int solve() {
  int n, p; cin >> n >> p;
  for (int i = 1; i <= n; ++i) {
    int a, b; cin >> a >> b;
    w[i] = {a, i}, w[n+i] = {b, i};
  }
  sort(w+1, w+2*n+1);

  map<int, int> mp1, mp2;
  int res = 0;
  for (int i = 1, j = 1; j <= 2*n; ++j) {
    mp1[w[j].id]++, mp2[w[j].id]++;
    while (i<=j && (LL)w[i].value*100<(LL)w[j].value*p) {
      mp1[w[i].id]--;
      if (mp1[w[i].id] == 0) mp1.erase(w[i].id);
      ++i;
    }
    if ((int)mp2.size() >= n) res = max(res, (int)mp1.size());
  }
  return res;
}
```

```
int main()
{
  ios::sync_with_stdio(false);
  cin.tie(0);
  int T; cin >> T;
  for (int i = 1; i <= T; ++i) cout << "Case #" << i << ": " << solve() << "\n";
  return 0;
}
```

CF335B Palindrome

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

using namespace std;

const int N = 5e4 + 5, M = 2600 + 5;
char s[N];
int f[M][M];

string dfs(int l, int r, int len) {
  if (len == 0) return "";
  if (len == 1) { string t; t += s[l]; return t; }


  if (s[l]==s[r] && f[l][r]==f[l+1][r-1]+2) return s[l] + dfs(l+1,r-1,len-2) +
s[r];
  if (f[l][r] == f[l][r-1]) return dfs(l,r-1,len);
  return dfs(l+1,r,len);
}

int main()
{
  cin >> (s+1);
  int n = strlen(s+1);
  if (n >= 2600) {
    map<char, int> mp;
    for (int i = 1; i <= n; ++i) mp[s[i]]++;
    for (char i = 'a'; i <= 'z'; ++i) {
      if (mp[i] >= 100) {
        for (int j = 1; j <= 100; ++j) cout << i;
        cout << endl;
        return 0;
      }
    }
  }

  for (int i = 1; i <= n; ++i) f[i][i] = 1;
  for (int len = 2; len <= n; ++len) {
    for (int i = 1; i+len-1 <= n; ++i) {
      int j = i + len - 1;
```

```
        f[i][j] = max(f[i][j-1], f[i+1][j]);
        if (s[i] == s[j]) f[i][j] = max(f[i][j], f[i+1][j-1]+2);
      }
    }

    cout << dfs(1, n, min(100, f[1][n])) << endl;
    return 0;
}
```