综合练习

人员

王毅博、阮文璋、褚锦轩、王承周 到课

上周作业检查

上周作业链接: https://cppoj.kids123code.com/contest/755

2025-0907 周日13:30 (逆元)

刷新

#	用户名	姓名	编程分	时间	A	B	C	D	E	F
1	wangchengzhou	王承周	600	10110	100	100	100	100	100	100
2	dongyuhan	董昱含	537	2526	100	100	100	100	100	37
3	ruanwenzhang	阮文璋	500	2083	100	100	100	100	100	
4	wangyibo	王毅博	500	10118	100	100	100	100		100
5	chujinxuan	褚锦轩	400	1644	100	100	100	100		

本周作业

https://cppoj.kids123code.com/contest/794 (课上讲了 A ~ E 题, 课后作业是 F 题)

课堂表现

今天上课先做了一道完全背包的题目, 能看出全班大部分同学把之前背包的内容已经遗忘很多了, 同学们课下要再去复习复习之前学的背包题目。

课堂内容

A Piece of Cake (上周作业)

map 套 pair 维护

判断一个草莓在哪一块蛋糕上, 这个过程可以用 二分查找 logn 来实现, 然后用 map 套 pair 记录这块蛋糕上草莓数量 +1

```
#include <bits/stdc++.h>
#define x first
#define y second

using namespace std;

typedef pair<int,int> PII;
const int maxn = 2e5 + 5;
int a[maxn], b[maxn];
struct node {
  int x, y;
```

```
} w[maxn];

int main()
{
  int W, H; cin >> W >> H;
  int T; cin >> T;
  for (int i = 1; i <= T; ++i) cin >> w[i].x >> w[i].y;

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

  map<PII, int> mp;
  for (int i = 1; i <= T; ++i) {
    int x = w[i].x, y = w[i].y;
    int tx = lower_bound(a+1, a+n+1, x) - a, ty = lower_bound(b+1, b+m+1, y) - b;
    mp[{tx,ty}]++;
  }

  int minn = 1e9, maxx = 0;
  if (mp.size() != (n+1)*(m+1)) minn = 0;
  for (auto it : mp) minn = min(minn, it.y), maxx = max(maxx, it.y);
  cout << minn << " " << maxx << endl;
  return 0;
}
```

疯狂的采药

完全背包 模板题

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

using namespace std;

typedef long long LL;
const int maxn = 1e7 + 5;
LL f[maxn];

int main()
{
  int T, n; cin >> T >> n;
  while (n -- ) {
    int a, b; cin >> a >> b;
    for (int i = a; i <= T; ++i) f[i] = max(f[i], f[i-a]+b);
  }
  cout << f[T] << endl;
  return 0;
}
```

纸币问题 3

完全背包 模板题

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

using namespace std;

const int maxn = 1e4 + 5;
const int mod = 1e9 + 7;
int f[maxn];

int main()
{
    int n, m; cin >> n >> m;
    f[0] = 1;
    for (int i = 1; i <= n; ++i) {
        int x; cin >> x;
        for (int j = x; j <= m; ++j) f[j] = (f[j] + f[j-x]) % mod;
    }
    cout << f[m] << endl;
    return 0;
}
```

纸币问题 2

先枚举背包体积, 在枚举纸币的类别即可

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

using namespace std;

const int maxn = 1e4 + 5;
const int mod = 1e9 + 7;
int w[maxn], f[maxn];

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

    f[0] = 1;
    for (int i = 1; i <= m; ++i) {
        for (int j = 1; j <= n; ++j) {
            if (i >= w[j]) f[i] = (f[i] + f[i-w[j]]) % mod;
        }
    }
    cout << f[m] << endl;
```

```
    return 0;
}
```

Colorful Blocks

题目要求最多有 k 对相邻对的颜色相同, 因此可以从 0~n 枚举相邻对的数量

若有 i 对相邻, 那么有 C(n-1,i) * m * (m-1)^(n-i-1) 种方案

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

using namespace std;

typedef long long LL;
const int maxn = 2e5 + 5;
const int mod = 998244353;
int fac[maxn], i_fac[maxn];

int qmod(int a, int k) {
  int res = 1;
  while (k) {
    if (k&1) res = (LL)res * a % mod;
    a = (LL)a * a % mod;
    k >>= 1;
  }
  return res;
}

void init() {
  fac[0] = i_fac[0] = 1;
  for (int i = 1; i < maxn; ++i) fac[i] = (LL)fac[i-1]*i % mod;
  for (int i = 1; i < maxn; ++i) i_fac[i] = qmod(fac[i], mod-2);
}

int C(int n, int m) {
  return (LL)fac[n] * i_fac[m] % mod * i_fac[n-m] % mod;
}

int main()
{
  init();

  int n, m, k; cin >> n >> m >> k;
  int res = 0;
  for (int i = 0; i <= k; ++i) {
    int t = (LL)C(n-1,i) * m % mod * qmod(m-1, n-i-1) % mod;
    res = (res + t) % mod;
  }
  cout << res << endl;
```

```
  return 0;
}
```

Count Bracket Sequences

dp, 设 f[i][j] 代表 以第 i 位结尾, 1~i 中 (-) ==j 时的方案数

答案是 f[n][0], 初值是 f[0][0] = 1

转移分为 左括号转移 和 右括号转移 两种情况

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

using namespace std;

typedef long long LL;
const int maxn = 3000 + 5;
const int mod = 998244353;
char s[maxn];
int f[maxn][maxn];  // f[i][j]: 以第 i 位结尾, 1~i 中 ( - ) ==j 时的方案数

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

  f[0][0] = 1;
  for (int i = 1; i <= n; ++i) {
    for (int j = 0; j <= i; ++j) {
      if (s[i]=='(' || s[i]=='?') {
        if (j > 0) f[i][j] = f[i-1][j-1];
      }
      if (s[i]==')' || s[i]=='?') {
        f[i][j] = (f[i][j] + f[i-1][j+1]) % mod;
      }
    }
  }

  if (s[n] == '(') cout << 0 << endl;
  else cout << f[n][0] << endl;
  return 0;
}
```