dp入门

人员

崔宸赫、于家瑞、郭栩睿、陶汇笙、洪晨栋、洪晨棋、于霄龙 到课, 王恩泽 线上

上周作业检查

上周作业链接: https://cppoj.kids123code.com/contest/1338

#	用户名	姓名	编程分	时间	A	B	C	D	E
1	taohuisheng	陶汇笙	260	2653	100	100	20	40	
2	guoxurui	郭栩睿	232	2646	100	100	20	12	
3	yujiarui	于家瑞	220	2324	100	100	20		
4	cuichenhe	崔宸赫	200	2638	100	100			

本周作业

https://cppoj.kids123code.com/contest/1476 (课上讲了 A ~ F 这些题, 课后作业是 G 题)

课堂表现

今天讲了 dp 这个知识点, 同学们课上整体吸收的还不错, dp 一定要熟记课上的四步走来做题。

课堂内容

dp 四步走

1. 定义状态
2. 考虑如何用状态求答案
3. 设计状态转移
4. dp赋初值

平铺图案 (上周作业)

二维前缀和直接维护即可

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

using namespace std;
```

```cpp
const int maxn = 1000 + 5;
char s[maxn][maxn];
int p[maxn][maxn];

int calc(int n, int x, int y) {
  int c1 = x / n, c2 = y / n;
  int sx = x % n, sy = y % n;
  int res = c1 * c2 * p[n][n] + c1 * p[n][sy] + c2 * p[sx][n] + p[sx][sy];
  return res;
}

signed main()
{
  int n, m; cin >> n >> m;
  for (int i = 1; i <= n; ++i) cin >> (s[i]+1);
  for (int i = 1; i <= n; ++i) {
    for (int j = 1; j <= n; ++j) {
      p[i][j] = p[i-1][j] + p[i][j-1] - p[i-1][j-1] + (s[i][j]=='B');
    }
  }

  while (m -- ) {
    int a, b, c, d; cin >> a >> b >> c >> d;
    ++a, ++b, ++c, ++d;
//    cout << "------------------ ";
    cout << calc(n,c,d) - calc(n,a-1,d) - calc(n,c,b-1) + calc(n,a-1,b-1) << endl;
  }
  return 0;
}
```

[GESP样题 六级] 下楼梯

递推方程: f[i] = f[i-3] + f[i-2] + f[i-1]

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

using namespace std;

typedef long long LL;
const int maxn = 60 + 5;
LL f[maxn];

int main()
{
  f[1] = 1, f[2] = 2, f[3] = 4;
  int n; cin >> n;
  for (int i = 4; i <= n; ++i) f[i] = f[i-3] + f[i-2] + f[i-1];
  cout << f[n] << endl;
  return 0;
}
```

[蓝桥杯 2023 省 Python B] 松散子序列

定义 f[i]: 以 i 结尾并选第 i 个数的时候, 最大能获得的价值是多少

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

using namespace std;

const int maxn = 1e6 + 5;
char s[maxn];
int f[maxn];

int main()
{
  cin >> (s+1);
  int n = strlen(s+1);
  for (int i = 1; i <= n; ++i) {
    if (i <= 2) f[i] = s[i]-'a'+1;
    else f[i] = max(f[i-2], f[i-3]) + s[i]-'a'+1;
  }
  cout << max(f[n-1], f[n]) << endl;
  return 0;
}
```

最长上升子序列

f[i]: 以第 i 个数结尾并选第 i 个数的时候, 能得到的最长上升子序列长度是多少

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

using namespace std;

const int maxn = 5000 + 5;
int w[maxn], f[maxn];

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

    for (int i = 1; i <= n; ++i) {
    f[i] = 1;
        for (int j = 1; j < i; ++j) {
            if (w[j] < w[i]) f[i] = max(f[i], f[j]+1);
        }
    }

    int res = 0;
    for (int i = 1; i <= n; ++i) res = max(res, f[i]);
    cout << res << endl;
```

```
    return 0;
}
```

P8707 [蓝桥杯 2020 省 AB1] 走方格

定义 f[i][j]: 走到 (i,j) 这个点时, 共有多少不同的方案

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

using namespace std;

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

int main()
{
  int n, m; cin >> n >> m;
  f[1][1] = 1;
  for (int i = 1; i <= n; ++i) {
    for (int j = 1; j <= m; ++j) {
      if (i==1 && j==1) continue;
      if (i%2==0 && j%2==0) continue;
      f[i][j] = f[i-1][j] + f[i][j-1];
    }
  }
  cout << f[n][m] << endl;
  return 0;
}
```

[IOI 1994] 数字三角形 Number Triangles

f[i][j]: 到 (i,j) 这个点最多能得多少分

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

using namespace std;

const int maxn = 1000 + 5;
int f[maxn][maxn];

int main()
{
  int n; cin >> n;
  for (int i = 1; i <= n; ++i) {
    for (int j = 1; j <= i; ++j) {
      int x; cin >> x;
      f[i][j] = max(f[i-1][j-1], f[i-1][j]) + x;
    }
  }
```

```
  int res = 0;
  for (int i = 1; i <= n; ++i) res = max(res, f[n][i]);
  cout << res << endl;
  return 0;
}
```

三倍经验

f[i][j][k]: 到 (i,j) 这个点, 如果只进行 k 次乘 3 操作, 最多能得多少分

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

using namespace std;

typedef long long LL;
const int maxn = 100 + 5;
const LL inf = 0x3f3f3f3f3f3f3f3f;
LL f[maxn][maxn][maxn];

int main()
{
  int n, m; cin >> n >> m;
  m = min(m, n);

  memset(f, -0x3f, sizeof(f)); f[0][0][0] = 0;
  for (int i = 1; i <= n; ++i) {
    for (int j = 1; j <= i; ++j) {
      LL x; cin >> x;
      for (int k = 0; k <= m; ++k) {
        if (k >= 1) f[i][j][k] = max(f[i-1][j-1][k-1], f[i-1][j][k-1]) + x*3;
        f[i][j][k] = max(f[i][j][k], max(f[i-1][j-1][k], f[i-1][j][k]) + x);
      }
    }
  }

  LL res = -inf;
  for (int i = 1; i <= n; ++i) {
    for (int j = 0; j <= m; ++j) res = max(res, f[n][i][j]);
  }
  cout << res << endl;
  return 0;
}
```