dp综合练习

人员

于家瑞、郭栩睿、洪晨栋、洪晨棋、崔宸赫、王恩泽 到课, 于霄龙 线上

上周作业检查

上周作业链接: https://cppoj.kids123code.com/contest/2166

#	用户名	姓名	编程分	时间	A	B	C	D	E	F
1	guoxurui	郭栩睿	600	590	100	100	100	100	100	100
2	wangenze	王恩泽	600	597	100	100	100	100	100	100
3	yujiarui	于家瑞	600	602	100	100	100	100	100	100
4	taohuisheng	陶汇笙	400	285	100	100	100	100		
5	yuxiaolong	于霄龙	300	239	100	100	100			

本周作业

https://cppoj.kids123code.com/contest/2235 (课上讲了 A ~ C 这些题, 课后作业是 D 题)

课堂表现

同学们今天听课做题整体都很认真, 希望同学们以后继续保持。

课堂内容

[USACO08DEC] Hay For Sale S (上周作业)

裸 01 背包

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

using namespace std;

const int maxn = 5e4 + 5;
bool f[maxn];

int main()
{
  int V, n; cin >> V >> n;
  f[0] = true;
  while (n -- ) {
    int v; cin >> v;
    for (int i = V; i >= v; --i) f[i] |= f[i-v];
```

```
  }
  for (int i = V; i >= 0; --i) {
    if (f[i]) { cout << i << endl; break; }
  }
  return 0;
}
```

[USACO3.1] 总分 Score Inflation (上周作业)

裸完全背包

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

using namespace std;

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

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

kkksc03考前临时抱佛脚

对 3 组分别做 01 背包, 看每一组能凑出来的最接近 sum/2 的数是多少

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

using namespace std;

int len[5];
const int maxn = 1200 + 5;
bool f[maxn];

int calc(int n) {
  memset(f, false, sizeof(f));
  f[0] = true;
  int sum = 0;
  while (n -- ) {
    int x; cin >> x; sum += x;
    for (int i = maxn-1; i >= x; --i) f[i] |= f[i-x];
  }
```

```
  for (int i = (sum+1)/2; i <= sum; ++i) {
    if (f[i]) return i;
  }
  return 0;
}

int main()
{
  for (int i = 1; i <= 4; ++i) cin >> len[i];

  int res = 0;
  for (int i = 1; i <= 4; ++i) res += calc(len[i]);
  cout << res << endl;
  return 0;
}
```

积木城堡

思想跟上道题一致, 一共分了 n 组, 分别对每一组做 01 背包的 dp, 记录每一组能凑出来哪些数, 最后找一个都能凑出来的最大值即可

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

using namespace std;

const int N = 10000 + 5, M = 100 + 5;
bool f[M][N];

void solve(int id) {
  f[id][0] = true;

  while (true) {
    int x; cin >> x;
    if (x == -1) break;
    for (int i = N-1; i >= x; --i) f[id][i] |= f[id][i-x];
  }
}

int main()
{
  int n; cin >> n;
  for (int i = 1; i <= n; ++i) solve(i);

  for (int i = N-1; i >= 0; --i) {
    bool flag = true;
    for (int j = 1; j <= n; ++j) {
      if (!f[j][i]) flag = false;
    }
    if (flag) { cout << i << endl; break; }
  }
```

```
  return 0;
}
```

[NOIP 2010 提高组] 乌龟棋

f[j1][j2][j3][j4]: 当使用了每种卡片分别为 j1, j2, j3, j4 张时, 能获得的最大价值是多少

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

using namespace std;

const int N = 120 + 5, M = 40 + 5;
int w[N], f[M][M][M][M];

int main()
{
  int n, m; cin >> n >> m;
  for (int i = 1; i <= n; ++i) cin >> w[i];
  int A = 0, B = 0, C = 0, D = 0;
  while (m -- ) {
    int x; cin >> x;
    if (x == 1) ++A;
    else if (x == 2) ++B;
    else if (x == 3) ++C;
    else ++D;
  }

  f[0][0][0][0] = w[1];
  for (int j1 = 0; j1 <= A; ++j1) {
    for (int j2 = 0; j2 <= B; ++j2) {
      for (int j3 = 0; j3 <= C; ++j3) {
        for (int j4 = 0; j4 <= D; ++j4) {
          if (!j1 && !j2 && !j3 && !j4) continue;

          int a = 0, b = 0, c = 0, d = 0;
          if (j1>=1) a = f[j1-1][j2][j3][j4];
          if (j2>=1) b = f[j1][j2-1][j3][j4];
          if (j3>=1) c = f[j1][j2][j3-1][j4];
          if (j4>=1) d = f[j1][j2][j3][j4-1];
          int pos = j1 + 2*j2 + 3*j3 + 4*j4 + 1;
          f[j1][j2][j3][j4] = max({a,b,c,d}) + w[pos];
        }
      }
    }
  }
  cout << f[A][B][C][D] << endl;
  return 0;
}
```