欢迎各位大佬乘坐东方特快列车（危）

Q1：01 背包

这个题目真的没有什么好讲的，我就想直接代码 CV 上来了。

就是记得第二层循环要逆向循环就好了，没有什么难度，送分题。

```cpp
#include <cstdio>
#define max(a, b) ((a) > (b) ? (a) : (b))
int n, m, money[1010];
long long val[1010], dp[200000];
int main() {
  scanf("%d%d", &n, &m);
  for (int i = 1; i <= n; i++) scanf("%d%lld", &money[i], &val[i]);
  for (int i = 1; i <= n; i++)
    for (int j = m; j >= money[i]; j--)
      dp[j] = max(dp[j - money[i]] + val[i], dp[j]);
  printf("%lld", dp[m]);
}
```

Q2：并查集

这个之前是带删除的，后来我觉得会不会太难了，所以就把带删除的删掉了，结果这题又变成了送分题，这是我没有想到的。

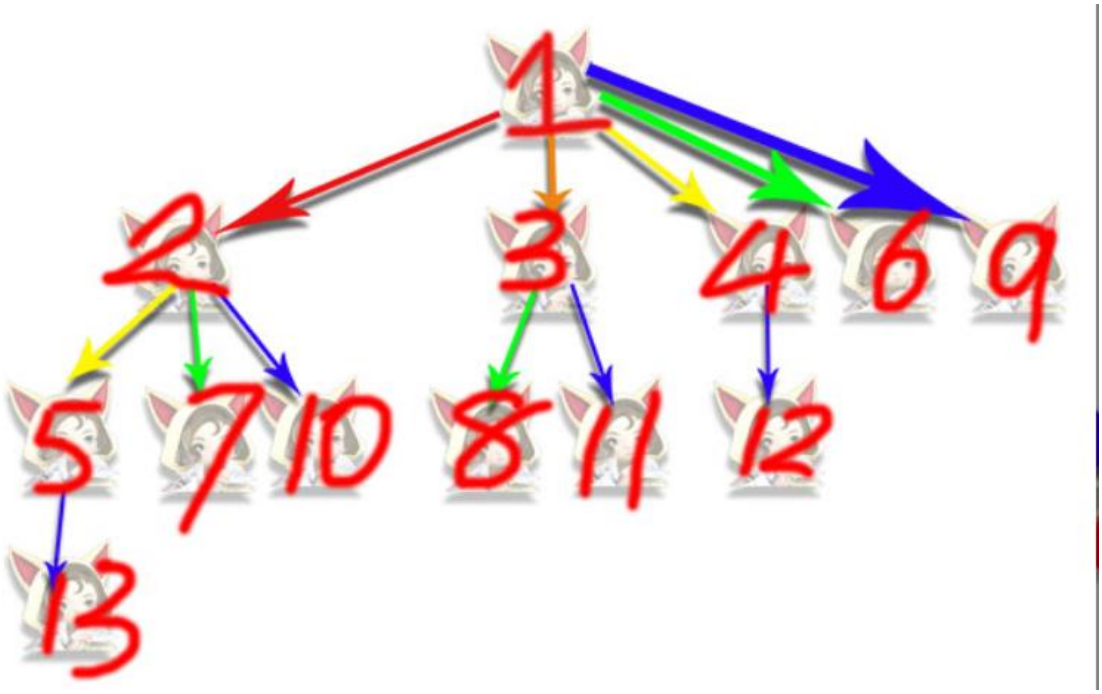
就是路径压缩记得写，跟普通并查集不一样的地方，就是用一个 size 数组记录一下这个人所在的团队有多少人，每次只要搜索到最高的那个祖先就可以知道这个人所在的团队有多少人了。是不是很简单。（不会真的有人循环过去找一遍吧。）

```cpp
#include <cstdio>
#define MAXN 300100
int n, m, tot, root[MAXN], size[MAXN];
int Find(int r) { return r != root[r] ? root[r] = Find(root[r]) : ro
ot[r]; }
int main() {
  scanf("%d%d", &n, &m);
  tot = n;
  for (int i = 1; i <= n; i++) root[i] = i, size[i] = 1;
  int t, u, v;
  for (int i = 1; i <= m; i++) {
    scanf("%d", &t);
    if (t == 1) {
      scanf("%d%d", &u, &v);
      int ru = Find(u), rv = Find(v);
      if (ru != rv) {
        tot--;
        root[ru] = rv;
        size[rv] += size[ru];
        size[ru] = 0;
      }
    }
    if (t == 2) {
      scanf("%d%d", &u, &v);
      puts(Find(u) == Find(v) ? "Yes" : "No");
    }
    if (t == 3) {
      scanf("%d", &u);
      printf("%d\n", size[Find(u)]);
    }
    if (t == 4) printf("%d\n", tot);
  }
}
```

Q3：这是一题思维题， 其实大家第一眼看过来会感觉“这就是 LCA 问题嘛！看我随便写一个模板 A 了。”然后发现这个建出来的树有点大。

就是一个找规律的题，我之前甚至在想，这个难度会不会比前面的简单，只是因为剧情的原因放在最后面，结果发现还是有点思维量的，这个跟 CF 上面的 DIV2 的 B~C 的难度感觉差不多。

我简单的讲一下这个怎么做，这个图稍微观察一下



很明显 可以发现这除了根节点之外所有点的编号 减去他的父亲节点的编号 都可以在斐波那契数列上面找到。

这个很容易证明，按照创建分身的时间来算，

time = 1 的时候第 2 号分身出来 = 1

time=2 的时候第 3 号分身出来 = 1

time = 3 的时候刚好是 2 号分身召唤分身的时候也就是 4 号和 5 号分身 = 2

time = 4 的时候是 6 号 7 号 8 号， = 3

然后把 time 建立成数列就是斐波那契数列了。

而根据这个思想 1 号分身的直接儿子是不是（1+1） （1+2） （1+3）（1+5）（1+8）

下面的也都可以这样同理获得。

显而易见如果想要知道他的父亲是谁，只要减去他能够减去的最大的斐波那契数，就可以获得他的父亲，而每次只要对两个数大的进行操作，直到两个数相等，那么相等的时候就是答案了。

```
#include <algorithm>
#include <cstdio>
long long fib[100], n, a, b;
int main() {
  fib[1] = 1, fib[2] = 2;
  for (int i = 3; fib[i - 1] < 1e15; i++) fib[i] = fib[i - 1] + fib[
i - 2];
  scanf("%lld", &n);
  for (int i = 1; i <= n; i++) {
    scanf("%lld%lld", &a, &b);
    while (a != b) {
      if (a > b)
        a -= fib[std::lower_bound(fib + 1, fib + 63, a) - fib - 1];
      else
        b -= fib[std::lower_bound(fib + 1, fib + 63, b) - fib - 1];
    }
    printf("%lld\n", a);
  }
}
```

代码意外的短 #滑稽