

E2 -C problem

题目描述

小水獭在预推免面试过程中被问到了一个难题，想要紧急求助你，你能帮帮它嘛？

具体来说，给定正整数 n 和 m ，你需要构造 m 个非空正整数集合 S_1, S_2, \dots, S_m 满足：

- $S_i \subseteq \{1, 2, \dots, n\}$ 对 $1 \leq i \leq m$ 成立。
- $S_i \not\subseteq S_j$ 对 $1 \leq i, j \leq m, i \neq j$ 成立。

当然，题目有可能是无解的，此时你需要告知小水獭。

题解思路

- 由分析，为了避免某个集合是另外一个集合的子集，当 m 个非空正数集合中元素个数相同时， m 能够取到最大值，且最大值为组合数 C_n^k 。即只要两集合中元素不完全相同，即符合题意。
- 对于 n 取 $[1, 10]$ 可知 k 取如下值可以使得组合数有最大值：

$$C_1^1, C_2^1, C_3^2, C_4^2, C_5^2, C_6^3, C_7^3, C_8^4, C_9^4, C_{10}^5$$

- 因此对于输入的 n ，首先判断 m 是否超过组合数最大值，若超过则返回 -1，否则按照每个集合有 k 个元素， k 重循环输出结果即可。

代码

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

using namespace std;

int main() {
    int t;
    scanf("%d", &t);
    while (t--) {
        int n, m;
        scanf("%d %d", &n, &m);
        if (m == 1) {
            printf("1 1\n");
        }
        else if (m == 2 && n == 2) {
            printf("1 1\n");
            printf("1 2\n");
        }
        else if (n >= 3 && n <= 5) {
            if (m > n * (n-1) / 2) {
                printf("-1\n");
                continue;
            }
        }
        int count = 0;
```

```

bool flag = false;
for (int i = 1; i < n; i++) {
    for (int j = i + 1; j <= n; j++) {
        printf("2 %d %d\n", i, j);
        count++;
        if (count == m) {
            flag = true;
            break;
        }
    }
    if (flag) break;
}

else if (n >= 6 && n <= 7) {
    if (m > n * (n-1) * (n-2) / 6) {
        printf("-1\n");
        continue;
    }
    int count = 0;
    bool flag = false;
    for (int i = 1; i <= n - 2; i++) {
        for (int j = i + 1; j <= n - 1; j++) {
            for (int k = j + 1; k <= n; k++) {
                printf("3 %d %d %d\n", i, j, k);
                count++;
                if (count == m) {
                    flag = true;
                    break;
                }
            }
            if (flag) break;
        }
        if (flag) break;
    }
}

else if (n >= 8 && n <= 9) {
    if (m > n * (n-1) * (n-2) * (n-3) / 24) {
        printf("-1\n");
        continue;
    }
    int count = 0;
    bool flag = false;
    for (int i = 1; i <= n - 3; i++) {
        for (int j = i + 1; j <= n - 2; j++) {
            for (int k = j + 1; k <= n - 1; k++) {
                for (int p = k + 1; p <= n; p++) {
                    printf("4 %d %d %d %d\n", i, j, k, p);
                    count++;
                    if (count == m) {
                        flag = true;
                        break;
                    }
                }
            }
            if (flag) break;
        }
    }
}

```

```

        if (flag) break;
    }
    if (flag) break;
}
}
else if (n == 10){
    if (m > n * (n-1) * (n-2) * (n-3) * (n-4) / 120) {
        printf("-1\n");
        continue;
    }
    int count = 0;
    bool flag = false;
    for (int i = 1; i <= n - 4; i++) {
        for (int j = i + 1; j <= n - 3; j++) {
            for (int k = j + 1; k <= n - 2; k++) {
                for (int p = k + 1; p <= n - 1; p++) {
                    for (int q = p + 1; q <= n; q++) {
                        printf("5 %d %d %d %d %d\n", i, j, k, p, q);
                        count++;
                        if (count == m) {
                            flag = true;
                            break;
                        }
                    }
                }
            }
            if (flag) break;
        }
        if (flag) break;
    }
    if (flag) break;
}
}
}
else {
    printf("-1\n");
}
}
return 0;
}

```