

P1036 [NOIP2002 普及组] 选数

题目描述

已知  $n$  个整数  $x_1, x_2, \dots, x_n$ ，以及 1 个整数  $k$  ( $k < n$ )。从  $n$  个整数中任选  $k$  个整数相加，可分别得到一系列的和。例如当  $n=4$ ， $k=3$ ，4 个整数分别为 3, 7, 12, 19 时，可得全部的组合与它们的和为：

$$3+7+12=22$$

$$3+7+19=29$$

$$7+12+19=38$$

$$3+12+19=34$$

现在，要求你计算出和为素数共有多少种。

例如上例，只有一种的和为素数：3+7+19=29。

输入格式

第一行两个空格隔开的整数  $n, k$  ( $1 \leq n \leq 20, k < n$ )。

第二行  $n$  个整数，分别为  $x_1, x_2, \dots, x_n$  ( $1 \leq x_i \leq 5 \times 10^6$ )。

输出格式

输出一个整数，表示种类数。

输入样例

```
4 3
3 7 12 19
```

输出样例

```
1
```

解析

本题的难点在于如何去重，在这里我们使用一个不降原则来解决这个问题。

不降原则是什么意思呢？我们用一个简单的例子来进行理解：从4个数当中选择3个。

很明显，第一波被选出来的一定是：1,2,3，那么接下来的是什么呢？

3之后还有4，那么结果就是1,2,4。

至此，第3位就全部选择完毕，我们需要来调整第2位，很明显只有一个结果：1,3,4

最后，我们来调整第1位，结果明显为：2,3,4

大家看懂了吧，每一位的数字只能向上增长，这样就保证了我们永远不会遇到重复的数字。这就是不降原则。

## 编码

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

using namespace std;

int ans = 0, n, k, a[20];

//素数判断
bool prime(int x) {
    int i;
    for (i = 2; i <= floor(sqrt(x)); i++) {
        if (x % i == 0) {
            return false;
        }
    }
    return true;
}

//深搜
//start: 开始选数的地方
//count: 已经选了几个数
//sum: 到目前为止选的数的和
void dfs(int start, int count, int sum) {
    //如果已经选了k个数，判断和是不是素数
    //ans是符合条件的和的个数
    if (count == k && prime(sum)) {
        ans++;
        return;
    }
    //从开始选数的地方到n
    //每重循环都是一种可能性
    //例如：选第二个数，start=2,n=4,有三种可能性(2,3,4)，调用自身3次
    for (int i = start; i <= n; i++) {
        //参数1: 从a里面下一个数开始选
```

```

        //参数2: 已经选的数的个数+1
        //参数3: sum加上这次选的数

        //因为直接从下一个数开始选, 所以不可能选到之前选过的数
        //无需判断当前的数是否被选过
        dfs(i + 1, count + 1, sum + a[i]);
    }
}

int main() {
    //读入基础数据
    int i;
    cin >> n >> k;
    for (i = 1; i <= n; i++) {
        cin >> a[i];
    }
    dfs(1, 0, 0);
    //从第一个数开始找, 已经找了0个数, 目前的和是0
    cout << ans << endl;
    return 0;
}

```

逻辑航线培优教育, 信息学奥赛培训专家。

扫码添加作者获取更多内容。

