

组合取数算法

1. 组合取数

【问题描述】

从 1, 2, …, N 这 N 个自然数中, 任取 R 个数进行组合, ($4 \leq N \leq 8$, $1 \leq R < N$ 。若 $R=N$?) 然后列出全部组合。

例如: 当 $N=5$, $R=3$ 时, 列出的全部组合如下:

```
1 2 3      1 2 4      1 2 5
1 3 4      1 3 5      1 4 5
2 3 4      2 3 5      2 4 5
3 4 5
S=10      C(5,3)
```

【程序说明】对任给出的 r 和 n, 则

第一个组合为 1, 2, …, r

然后变化第 r 位: 第二个组合为 1, 2, …, r+1

继续变化第 r 位, 可以变到 n; 然后再变化第 r-1 位, r-2 位, …, 直至第 1 位的变化。

用数组 b[20]记录组合, s 记录组合总数。

【程序清单】

```
#include <cstdio>
#include <iostream>
using namespace std;

int n, r, s, i, j;

int main()
{
    cin >> n >> r;
    int b[r+1];

    for (int i=0; i<=r; i++) b[i]=i;

    while (b[0]==0)
    {
        s++;
        for (int i=1; i<=r; i++) cout << b[i] << " ";
        cout << endl;

        j = r;
        while(b[j]==n-r+j) j--;
        //若允许 R=N, 则要改一下条件, 否则会出现问题
        b[j]++;
    }
}
```

```

        for (int i=j+1; i<=r; i++) b[i]=b[i-1]+1;
    }

    cout << s << endl;
    return 0;
}

```

2. 选数

【问题描述】

给定 n 个整数 x_1, x_2, \dots, x_n ($1 \leq x_i \leq 5000000$)，以及一个整数 k ($1 \leq n \leq 20, k \leq n$)，从 n 个整数中任选 k 个整数相加，可以分别得到一系列的和。例如，当 $n=4, k=3$ ，且四个整数分别为 3、7、12、19 时，可以得到的全部组合及它们的和为：

$$3+7+12=22 \quad 3+7+19=29 \quad 7+12+19=38 \quad 3+12+19=34$$

现在，要求你计算出和为素数的组合共有多少种。例如上例，只有一种的和为素数：3+7+19=29。

【输入】 键盘输入，格式为：

n, k
 x_1, x_2, \dots, x_n

【输出】 屏幕输出，格式为：

一个整数（满足条件的组合的种数）

【输入输出样例】

输入：

4 3
 3 7 12 19

输出：

1

【参考程序】

```

#include <cstdio>
#include <iostream>
using namespace std;

int main()
{
    int n, k, f, sum, i, j;
    cin >> n >> k;
    int a[n+1], b[k+1];

    for (i=1; i<=n; i++) cin >> a[i];

    for (i=0; i<=k; i++) b[i]=i; //0,1,2,3

    int total=0;

```

```

int is_prime(int x);

while(b[0]==0)
{
    sum=0;
    for (i=1; i<=k; i++)  sum +=a[b[i]];
    f = is_prime(sum);
    if (f) total++;
    j = k;
    while(b[j]==n-k+j) j--;
    b[j]++;
    for (i=j+1; i<=k; i++) b[i]=b[i-1]+1;
}

cout << total << endl;
return 0;
}

int is_prime(int x)
{
    int ans = 1;
    for (int i=2; i<=int(sqrt(x)); i++)
        if (x % i==0) {
            ans =0;
            break;
        }

    return ans;
}

```