

2.27 课堂笔记 1

课后作业：（周四晚之前交，代码+评测结果截图，发 qq 或微信都行）

1、互质组

给定 n 个正整数，将他们分组，使得每组中任意两个数互质（两个数的公约数只有 1）。至少要分多少个组？

输入：

第一行是一个正整数 n ($1 \leq n \leq 10$)；

第二行是 n 个不大于 10000 的正整数。

输出：

一个正整数，即最少需要的组数。

样例输入：

6

14 20 33 117 143 175

样例输出：

3

2、洛谷 P2036

3、洛谷 P1019（选做）

回溯

回溯算法实际上一个类似枚举的搜索尝试过程，主要是在搜索尝试过程中寻找问题的解，当发现已不满足求解条件或者达到目标还要求其他达到目标的方式时，就“回溯”返回，尝试别的路径。

回溯算法思路：

```
dfs(){  
    (剪枝/不满足条件时直接return)  
    if(达到目标){  
        输出/停止  
        return;  
    }  
    for(循环范围){  
        if(满足条件){  
            保存当前状态  
            dfs();  
            此dfs结束之后回退到上一步  
        }  
    }  
}
```

练习 1:

组合。

输入两个正整数 m 和 n ($20 \geq m \geq n > 0$)，要求 n 个正整数相加的和为 m ，输出满足这个条件的正整数组合有多少。

输入描述：分行输入 m 和 n ($20 \geq m \geq n > 0$)。

输出描述：输出满足这个条件的正整数组合有多少。

【样例输入】

4

8

【样例输出】

5

【上述输入输出样例的进一步解释】

用户输入的两个正整数，即样例输入为 4 和 8，满足条件的有： $5+1+1+1=8$ 、 $4+2+1+1=8$ 、 $3+3+1+1=8$ 、 $3+2+2+1=8$ 、 $2+2+2+2=8$ （每组组合都由 4 个正整数组成且 4 个正整数的和等于 8）所以样例输出为 5。

```

1  #include <bits/stdc++.h>
2  using namespace std;
3  int n, m, cnt;
4  int num[110];
5
6  void dfs(int cc, int sum){
7      if(cc == m + 1){
8          if(sum == n){
9              cnt++;
10             for(int i = 1 ; i <= m-1; i++){
11                 cout << num[i] << "+" ;
12             }
13             cout << num[m] << endl;
14         }
15         return;
16     }
17     for(int i = n-m+1; i>0; i--){
18         if(i <= num[cc-1]){ //降序, 保证不重复输出
19             num[cc] = i;
20             dfs(cc+1, sum+i);
21         }
22     }
23 }
24
25 int main(){
26     cin >> n >> m;
27     num[0]=n-m+1; //降序时num[0]初始化为较大值
28     dfs(1,0);
29     cout << cnt;
30     return 0;

```

练习 2: P2404

题目描述

展开

任何一个大于1的自然数n，总可以拆分成若干个小于n的自然数之和。现在给你一个自然数n，要求你求出n的拆分成一些数字的和。每个拆分后的序列中的数字从小到大排序。然后你需要输出这些序列，其中字典序小的序列需要优先输出。

输入格式

输入：待拆分的自然数n。

输出格式

输出：若干数的加法式子。

输入输出样例

输入 #1	复制	输出 #1	复制
7		1+1+1+1+1+1 1+1+1+1+1+2 1+1+1+1+3 1+1+1+2+2 1+1+1+4 1+1+2+3 1+1+5 1+2+2+2 1+2+4 1+3+3 1+6 2+2+3 2+5 3+4	

```
1  #include <bits/stdc++.h>
2  using namespace std;
3  int n;
4  int num[110];
5
6  void dfs(int cc, int sum){ //cc:计数 sum: 当前所选数字的和
7      if(cc > n+1){
8          return; //计数超过了最大值时, 直接返回
9      }
10     if(sum == n){ //判断和是否等于n, 此时计数一定满足要求
11         for(int i = 1; i<cc-1; i++){
12             cout << num[i] << "+";
13         }
14         cout << num[cc-1] << endl;
15         return;
16     }
17     for(int i=1; i<n; i++){
18         if(i >= num[cc-1]){ //升序输出, 保证不重复
19             num[cc]=i;
20             dfs(cc+1, sum+i);
21         }
22     }
23 }
24
25 int main(){
26     cin >> n;
27     dfs(1,0);
28     return 0;
29 }
```

练习 3: P1135

题目描述

 展开

呵呵，有一天我做了一个梦，梦见了一种很奇怪的电梯。大楼的每一层楼都可以停电梯，而且第 i 层楼（ $1 \leq i \leq N$ ）上有一个数字 K_i （ $0 \leq K_i \leq N$ ）。电梯只有四个按钮：开，关，上，下。上下的层数等于当前楼层上的那个数字。当然，如果不能满足要求，相应的按钮就会失灵。例如：3, 3, 1, 2, 5 代表了 K_i （ $K_1 = 3, K_2 = 3, \dots$ ），从 1 楼开始。在 1 楼，按“上”可以到 4 楼，按“下”是不起作用的，因为没有 -2 楼。那么，从 A 楼到 B 楼至少要按几次按钮呢？

输入格式

共二行。

第一行为三个用空格隔开的正整数，表示 N, A, B （ $1 \leq N \leq 200, 1 \leq A, B \leq N$ ）。

第二行为 N 个用空格隔开的非负整数，表示 K_i 。

输出格式

一行，即最少按键次数，若无法到达，则输出 -1。

输入输出样例

输入 #1

 复制

输出 #1

 复制

```
5 1 5
3 3 1 2 5
```

```
3
```

说明/提示

对于 100% 的数据， $1 \leq N \leq 200, 1 \leq A, B \leq N, 0 \leq K_i \leq N$ 。

```

1  #include <bits/stdc++.h>
2  using namespace std;
3  int n, a, b;
4  int num[210];
5  int ans = 2147483647;
6  int walk[210];
7  bool flag;
8
9  void dfs(int cc, int times){
10     if(times >= walk[cc]){ //按电梯次数超过之前的次数, 直接返回
11         return;
12     }
13     walk[cc]=times; //刷新最少次数
14     if(cc==b){
15         flag = true;
16         ans = min(ans,times);
17         return;
18     }
19     if(cc+num[cc] <= n && cc+num[cc] >=0){ //向上走
20         dfs(cc+num[cc], times+1);
21     }
22     if(cc-num[cc] <= n && cc-num[cc] >=0){ //向下走
23         dfs(cc-num[cc], times+1);
24     }
25 }
26
27 int main(){
28     cin >> n >> a >> b;
29     for(int i =1; i<= n;i++){
30         cin >> num[i];
31     }
32     memset(walk,0x3f,sizeof(walk)); //数组元素初始化为无穷大
33     dfs(a,0);
34     if(flag){
35         cout << ans;
36     }
37     else{
38         cout << -1;
39     }
40     return 0;
41 }

```


练习 4: P1036

题目描述

[展开](#)

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。

输入输出样例

输入 #1

[复制](#)

输出 #1

[复制](#)

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

```
1
```

```

2  using namespace std;
3  int n, m, cnt;
4  int num[110];
5
6  bool check(int x){ //素数的判断
7      for(int i = 2; i<=sqrt(x); i++){
8          if(x%i==0){
9              return false;
10         }
11     }
12     return true;
13 }
14
15 void dfs(int cc, int sum, int startX){
16     if(cc == m){
17         if(check(sum)){
18             cnt++;
19         }
20         return;
21     }
22     for(int i = startX; i<=n; i++){ //startX保证不重复取数
23         dfs(cc+1, sum+num[i], i+1);
24     }
25 }
26
27 int main(){
28     cin >> n >> m;
29     for(int i=1;i<=n;i++){
30         cin>> num[i];
31     }
32     dfs(0,0,1);
33     cout << cnt;
34     return 0;
35 }

```