刷题

DK

# 暴力枚举

## 1.0 概

循环枚举

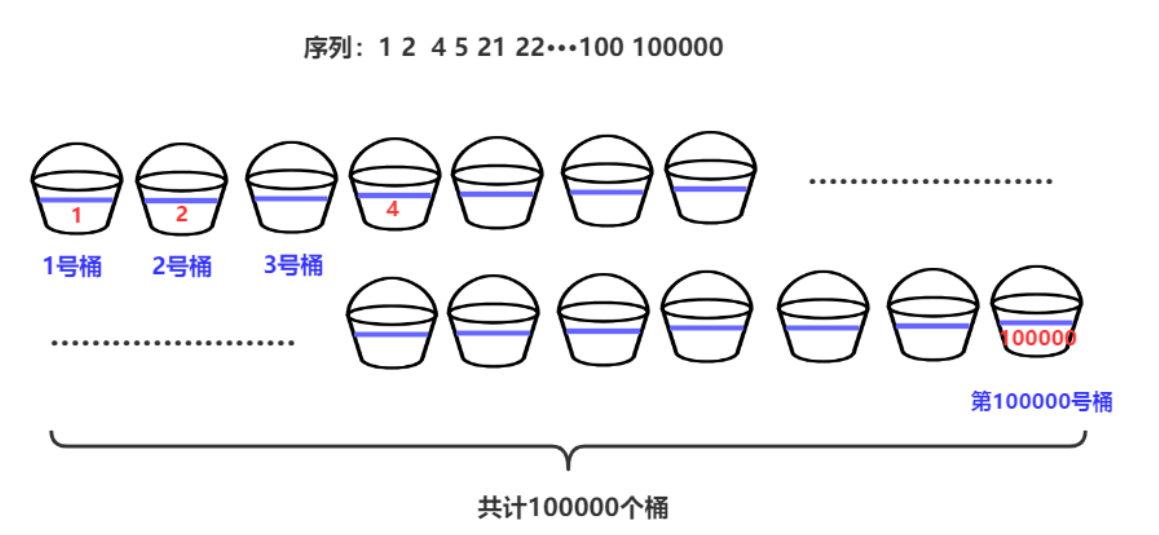
子集枚举

排列枚举

## 循环枚举

* **题目**

将 1, 2,\ldots, 91,2,…,9 共 99 个数分成三组，分别组成三个三位数，且使这三个三位数的比例是 A:B:C*A*:*B*:*C*，试求出所有满足条件的三个三位数，若无解，输出 No!!!。

* **涉及知识点**
* [桶](https://www.cnblogs.com/bigsai/p/13396391.html)
* [memset函数](https://www.runoob.com/cprogramming/c-function-memset.html)
* **代码**

1. // p1618 三连击升级版
2. #include <cstdio>
3. #include <iostream>
4. #include <algorithm>
5. #include <cstring>
6. using namespace std;
7. int b[10];
8. //桶
9. void go(int x) { //分解三位数到桶里
10. b[x % 10] = 1;
11. b[x / 10 % 10] = 1;
12. b[x / 100] = 1;
13. }
14. // 查验x,y,z三位数进入桶中之后是否完全占据九个数字
15. bool check(int x, int y, int z){
16. memset(b, 0 ,sizeof(b));
17. /\*memset函数\*/
18. if(y > 999 || z > 999) return 0;
19. go(x), go(y), go(z);
20. //查看b[1]~b[9]是否全为 1
21. for (int i = 1; i <= 9 ; ++i) {
22. if(!b[i]) return 0;
23. }
24. return 1;
25. }
26. int main(){
27. long long A,B,C,x,y,z,cnt = 0;
28. cin >> A >> B >> C;
29. //对x枚举
30. for(x = 123; x <= 987; x++){
31. / if(x \* B % A || x \* C % A ) continue; //?
32. y = x \* B / A, z = x \* C / A;
33. if(check(x,y,z))
34. printf("%lld %lld %lld\n",x,y,z),cnt++;
35. }
36. if(!cnt) puts("No!!!");
37. return 0;
38. }

## 子集枚举

* **题目**

P1036 [NOIP2002 普及组] 选数

* **涉及知识点**

\* 子集枚举 思路

用 a[30] 数组存储每个元素，用 U-1 表述全集(二进制代码，对每个集合进行编码)， 在分析子集和全集的关系时，可以用与或运算和异或运算。

（在c++中没有直接表示二进制数的方法，使用左移位）

例子：

若一个集合包含n个元素，那么可以用n位二进制代码表示这个全集

设n = 5，那么一共有5个元素，在全集中每个元素有两种可能(存在或不存在)，则有 00000 – 11111 的集合表示

仅包含第i个元素的集合 ： 1 << ( i - 1)

并集： A2 or A3

交集： a3 = a1 & a4

集合包含关系（判断A1包含A2）： ((a1 | a2) == a1 && (a1&a2) ==a2)

元素和集合的属于关系： 1<<(i-1) & a1

补集（和全集的异或关系）： a1 ^ a

* **代码**

1. #include <cstdio>
2. #include<bitset>
3. #include <stdio.h>
4. #include <string.h>
5. #include <iostream>
6. #include <iostream>
7. using namespace std;
8. /\*
9. \* 子集枚举 思路
10. \* 1 用 a[30] 数组存储每个元素
11. \* 2 用 U-1 表述全集(二进制代码，对每个集合进行编码)
12. \*/
13. int a[30]; //表示每个元素
14. /\*
15. \* check 质数
16. \*/
17. bool check(int x) {
18. for (int i = 2; i \* i <= x; ++i) {
19. if (x % i == 0) return 0;
20. }
21. return 1;
22. }
23. int main() {
24. int n, k, ans = 0;
25. /\* n: 整数数量 k: 选取整数个数\*/
26. cin >> n >> k; //先赋值n，再赋值k
27. for (int i = 0; i < n; ++i) {
28. scanf("%d", &a[i]);
29. }
30. int U = 1 << n; //z左移位->二进制数表示集合
31. /\*子集枚举： 对每个集合枚举\*/
32. for (int S = 0; S < U; S++) {
33. if (\_\_builtin\_popcount(S) == k) { //找到 k元子集合
34. int sum = 0;
35. for (int i = 0; i < n; ++i)
36. if (S & (1 << i))
37. sum += a[i]; /\* 第i个元素的左移位 1<<i 为 (1000...)b,和 S集合与运算，S集合的 第 i 位 如果为 1 则return 1 \*/
38. if (check(sum)) ans++;
39. }
40. }
41. cout << ans;
42. return 0;
43. }