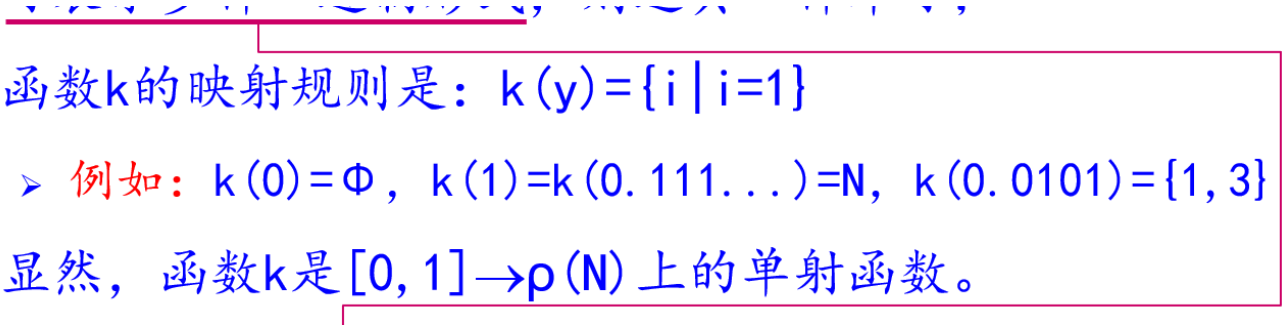
给出策略列举出包含n元素的集合的所有子集，并进行分析。

问题分析

我们都知道一个有n个元素的集合，共有2^n个集合，但是如何一个个不漏的将子集数出来输出是我们这个问题的重点。

在离散数学中，我们学过[0,1]到ρ(N)的映射k：



在这里实现了从二进制数到集合的映射

那么我们也可以将这种思想运用到本题中

算法思路

对于一个有n个元素的集合

我们创造一个n位的二进制数字

并将每个元素与二进制数字的一位建立对应关系

比如：{1，2，3}

000代表空集

001代表子集{1}

010代表子集{2}

只要我们能够按顺序的数二进制数

就可以按顺序的输出所有的子集

代码实现

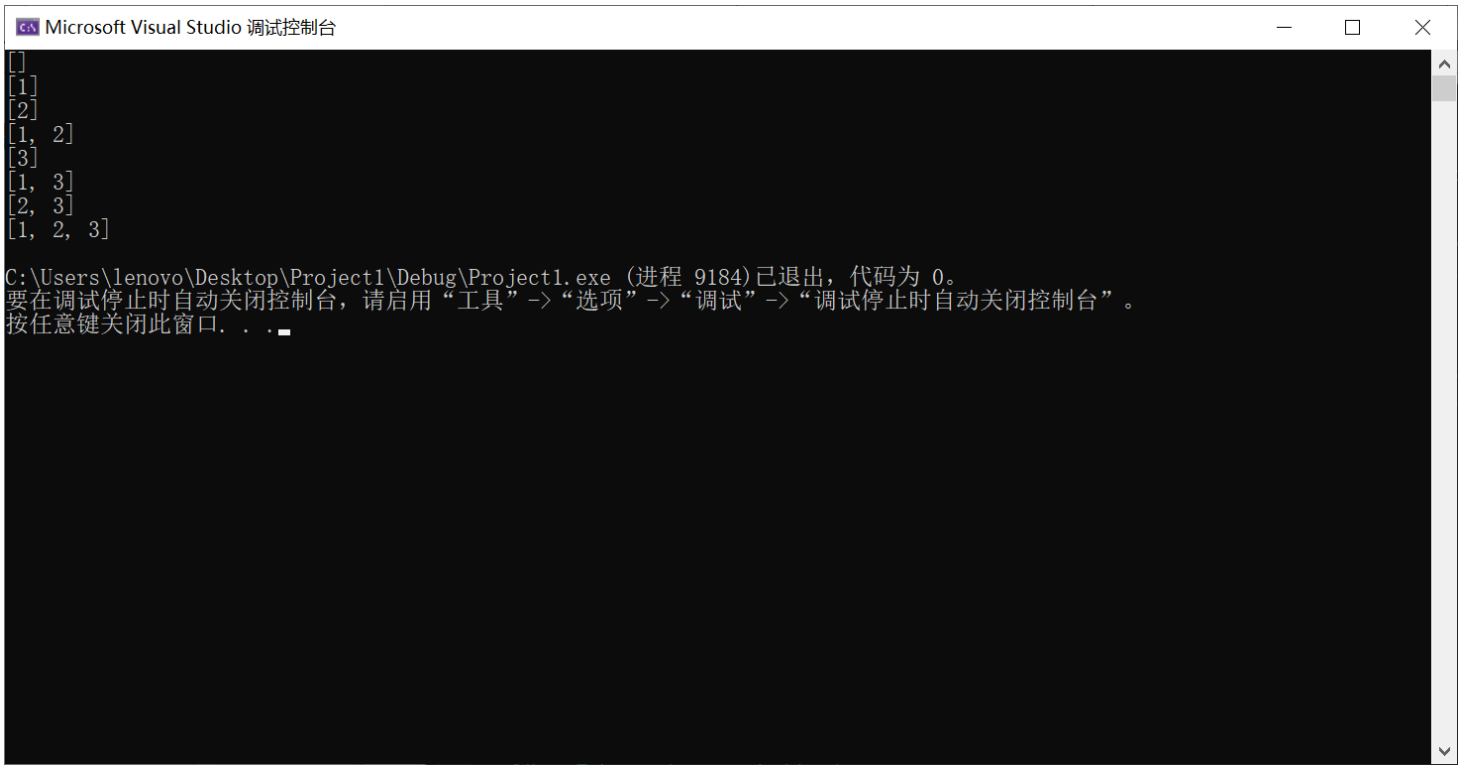
```cpp
#include <iostream>
#include <vector>
using namespace std;

void subsets(vector<int>& nums) {
    int n = nums.size();
    for (int i = 0; i < (1 << n); i++) { //对于2^n个集合依次遍历
        vector<int> subset;
        for (int j = 0; j < n; j++) {
            if (i & (1 << j)) { //判断第j位是不是1
                subset.push_back(nums[j]); //向subset内加入元素
            }
        }
        // 打印子集
        cout << "[";
        for (int k = 0; k < subset.size(); k++) {
            cout << subset[k];
            if (k != subset.size() - 1) cout << ", ";
        }
        cout << "]" << endl;
    }
}

int main() {
    vector<int> nums = { 1, 2, 3 };
    subsets(nums);
    return 0;
}
```

在这里用vector可以时间动态增添元素，如果使用数组需要提前定义好容量

结果展示



在机器学习、文本处理、推荐系统等应用领域，经常使用相似度指标，调研学习当前业界经常使用的相似度指标。

相似度指标

数据相似性一般使用距离来度量，这些距离包括欧氏距离 (Euclidean Distance)，曼哈顿距离 (Manhattan Distance)，切比雪夫距离 (Chebyshev Distance)，闵可夫斯基距离 (Minkowski Distance)，标准化欧氏距离 (Standardized Euclidean distance)，马氏距离 (Mahalanobis Distance)，夹角余弦 (Cosine)，皮尔逊相关系数（Pearson correlation），汉明距离 (Hamming distance)，杰卡德相似系数 (Jaccard similarity coefficient)，布雷柯蒂斯距离 (Bray Curtis Distance)等等

夹角余弦举例说明：

假设现在有两文档A和B，内容分别为：

A:我喜欢吃苹果，也喜欢吃香蕉

B:我不喜欢吃苹果，但是喜欢吃香蕉

根据关键字创建a，b向量

a = [1, 1, 1, 0]， b = [1, 0, 1, 1]

每个元素分别代表"我"、“喜欢”、“苹果”、"香蕉"这四个单词在文档中出现的次数

则cos(a,b) = (1×1+1×0+1×1+1×0)/(sqrt(3)*sqrt(3)) = 2/3

所以两篇文档的相似度为2/3