Subset

題目

Given an integer array `nums` of unique elements, return *all possible subsets (the power set)*.

The solution set must not contain duplicate subsets. Return the solution in any order.

Example 1:

```
Input: nums = [1,2,3]
Output:
[[],[1],[2],[1,2],[3],[1,3],[2,3],[1,2,3]]
```

圖一 樣本1輸入

Example 2:

```
Input: nums = [0]
Output: [[],[0]]
```

圖二 樣本2輸入

Constraints:

- `1 <= nums.length <= 10`
- `-10 <= nums[i] <= 10`
- All the numbers of `nums` are unique.

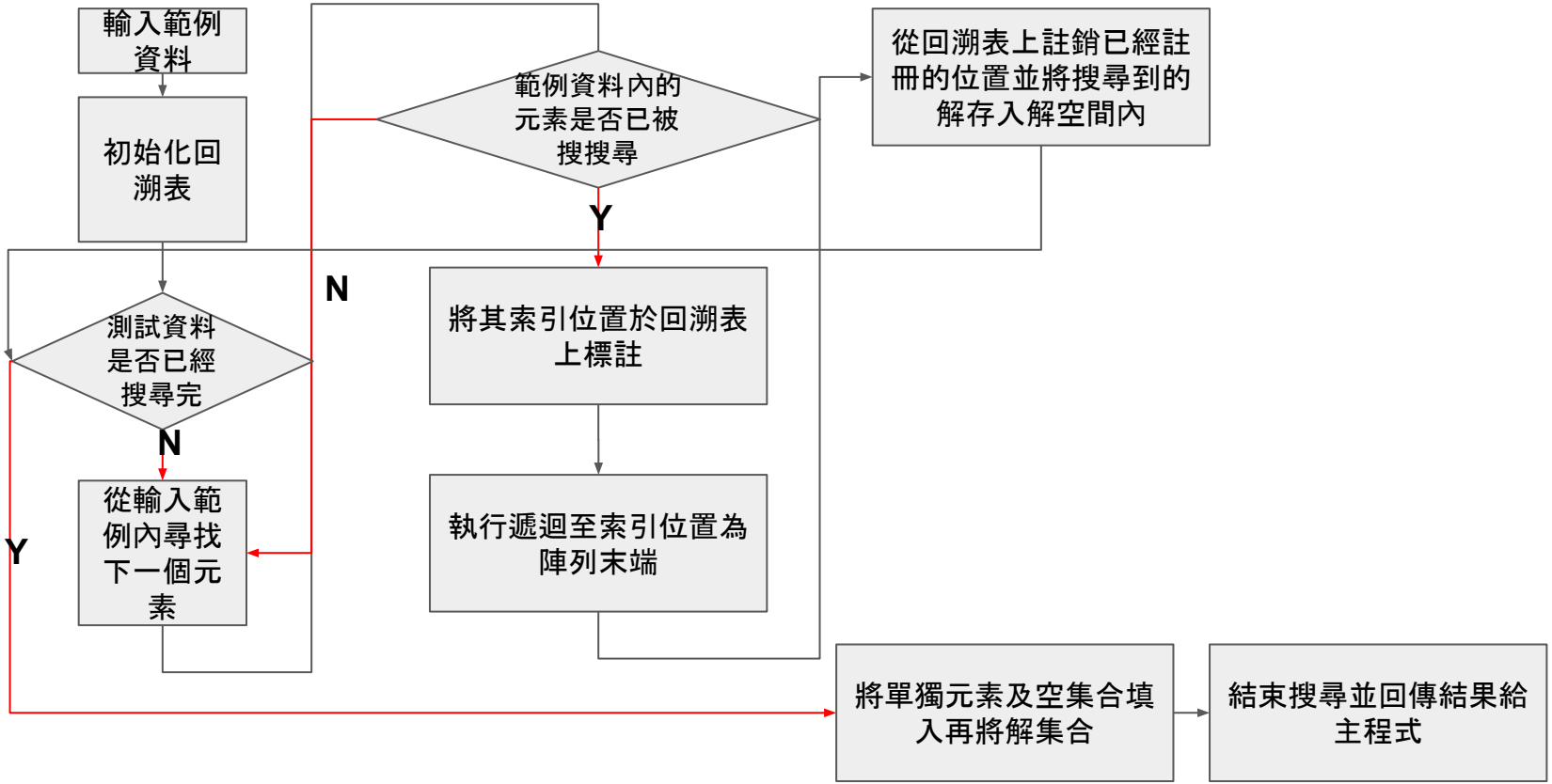
圖三 輸入限制

解題思路

1.依照題意上像是窮舉，故以backtracking上解決

2. 根據Leetcode上的提示，Output是List內的List，所以要回傳一個集合，且任兩個集合內的構成元素皆不相同。

程式邏輯


輸入範例資料
初始化回溯表
測試資料是否已經搜尋完
N
Y
從輸入範例內尋找下一個元素
範例資料內的元素是否已被搜搜尋
Y
N
將其索引位置於回溯表上標註
執行遞迴至索引位置為陣列末端
從回溯表上註銷已經註冊的位置並將搜尋到的解存入解空間內
將單獨元素及空集合填入再將解集合
結束搜尋並回傳結果給主程式

核心程式碼

```
  public List<List<Integer>> subsets(int[] nums) {
      Arrays.sort(nums);
      table=new int[nums.length];
      List<Integer> list=new ArrayList<>();
      List<List<Integer>> result=new
ArrayList<>();
      result.add(new ArrayList<>());
      doWork(nums,list,result);
      for(int i=0;i<nums.length;i++){
          list=new ArrayList<>();
          list.add(nums[i]);
          if (!result.contains(list)){
              result.add(list);
          }
      }

      return result;
  }
```

```
public void doWork(int[] nums,List<Integer>
list,List<List<Integer>>result){
      for(int i=0;i<nums.length;i++){
          if(table[i]==0) {
              table[i] = 1;
              list.add(nums[i]);
              doWork(nums, list,result);
              table[i] = 0;

              list.sort((Integer o1, Integer o2)->
o1-o2);
              if (!result.contains(list)){
                  result.add(list);
              }
              list=new ArrayList<>();
          }
      }
  }
```

程式碼詳見git

https://github.com/610621215/happygit123