

# 多个集合求组合，还是有点不一样的。

## 17.电话号码的字母组合

给定一个仅包含数字 2-9 的字符串，返回所有它能表示的字母组合。

给出数字到字母的映射如下（与电话按键相同）。注意 1 不对应任何字母。



▲ 17. 电话号码的字母组合

这个是一个组合问题即每一层都是数字对应的字符

假设输入的有k个数字，则满足条件的字符串的长度为k

1、回溯算法的参数和返回值

vector<string> result;

void backtracking(const string& digits,int index,string path)//index表示树的深度

2、回溯结束条件

if(index==digits.size()){回溯终止条件}

3、单层逻辑

int digit=digits[index]-'0';//将index指向的数字转为数字

string letters=letterMap[digit];

for(int i=0;i<letters.size();i++){

    backtracking(digits,index+1,path+letter[i]);

/\*\*

    path.push\_back(letter[i]);

    backtracking(digits,index+1,path);

    path.pop\_back();

\*/

}

## ☞ 输入1 \* #按键等等异常情况

代码中最好考虑这些异常情况，但题目的测试数据中应该没有异常情况的数据，所以我就没有加了。

「但是要知道会有这些异常，如果是现场面试中，一定要考虑到！」

## 如何实现数字和字母的映射？

对于该问题可以采用map或者定义一个数组(数组就是一个特殊的哈希表,其key值就是下标)  
string letterMap[10];

```
1  const string letterMap[10]={
2    "",//0
3    "",//1
4    "abc",//2
5    "def",//3
6    "ghi",//4
7    "jkl",//5
8    "mno",//6
9    "pqrs",//7
10   "tuv",//8
11   "wxyz",//9
12 }
```

## 最后的算法:

```
1  class Solution {
2  public:
3      const string letterMap[10]={
4          "",//0
5          "",//1
6          "abc",//2
7          "def",//3
8          "ghi",//4
9          "jkl",//5
10         "mno",//6
11         "pqrs",//7
12         "tuv",//8
13         "wxyz",//9
14     };
15     vector<string> result;
16     void backtracking(const string&digits,int depth,string path){
```

```

17  if(depth==(digits.size()-1){
18  result.push_back(path);
19  return ;
20  }
21  int digit=digits[depth]-'0';
22  string letter=letterMap[digit];
23  for(int i=0;i<letter.size();i++){
24  backtracking(digits,depth+1,path+letter[i]);
25  }
26  }
27  vector<string> letterCombinations(string digits) {
28  backtracking(digits,0,"");
29  return result;
30  }
31  };

```

注意：字符串string下标操作问题：

字符串虽然重载了赋值的运算符但是该运算符只是重载了一整个字符串赋值给另一个字符串  
string a,b;  
a[i]=b[i](×)  
a+=b[i];或者a.push\_back(b[i]);(✓)