

求集合的所有子集

现有一个包含N个元素的集合S，求集合S的所有子集？

例如：集合S包含三个元素{a, b, c}，则它的所有子集为：{}(空集), {a}, {b}, {c}, {a, b}, {a, c}, {b, c} 和{a, b, c}。

这里先用位操作的思路来求解，具体方法：用**2**进制**Bit**位来标记集合中的某个元素是否被选中，**1**代表选中，**0**代表未选中。例如集合{a, b, c}的所有子集可如下表示：

{ }(空集)	0 0 0
{a}	0 0 1
{b}	0 1 0
{c}	1 0 0
{a, b}	0 1 1
{a, c}	1 0 1
{b, c}	1 1 0
{a, b, c}	1 1 1

从上面的分析中也可以看出一个包含N个元素的集合S有 2^N 个子集，非常容易想到的方法就是遍历 $0 \sim 2^N - 1$ 的所有整数，并转化为二进制，按以上思路输出所有子集。但这里我们参照《求比正整数N大的最小正整数M，且M与N的二进制表示中有相同数目的1》的方法来求解集合S的所有子集。具体实现如下：

```
1. #include <iostream>
2. using namespace std;
3.
4. const int OK = 0;
5. const int ERROR = 1;
6.
7. int getNextN(int n)
8. {
9.     int temp1 = n & (-n);
10.    int temp2 = n + temp1;
11.    int ret = temp2 | ((n ^ temp2) / temp1) >> 2;
12.    return ret;
13. }
14.
```

```

15. template<class T>
16. void output(T set[], int k, int n)
17. {
18.     int index = 0;
19.     bool flag = false;
20.     cout << "{";
21.
22.     while (n)
23.     {
24.         if (n % 2 == 1)
25.         {
26.             if (flag)
27.             {
28.                 cout << ", ";
29.             }
30.             cout << set[k + index];
31.             flag = true;
32.         }
33.         n /= 2;
34.         index++;
35.     }
36.     cout << "}" << endl;
37. }
38.
39.
40. template<class T>
41. int SubSet(T set[], int k, int m)
42. {
43.     if (k < 0 || m < 0 || k > m)
44.     {
45.         return ERROR;
46.     }
47.
48.     output(set, k, 0);
49.     int num = m - k + 1;
50.     for (int i = 1; i <= num; i++)
51.     {
52.         int beginIdx = (1 << i) - 1;
53.         int endIdx = (1 << num) - (1 << (num - i));
54.         for (int j = beginIdx; j <= endIdx; j++)
55.         {
56.             output(set, k, j);
57.             j = getNextN(j);
58.         }

```

```

59.     }
60.
61.     return OK;
62. }
63.
64. int main(int argc, char *argv[])
65. {
66.     char str[10];
67.     cout << "请输入一个字符数组:";
68.     cin >> str;
69.     SubSet(str, 0, strlen(str) - 1);
70.     system("PAUSE");
71.     return EXIT_SUCCESS;
72. }

```

这里再提供一种递归的思路来解决问题，会借助一个全局变量来记录所有选中的元素，具体代码如下：

```

1. #include <iostream>
2. using namespace std;
3.
4. int g_used[10] = {0};
5.
6. template<class T>
7. void Output(T set[])
8. {
9.     bool flag = false;
10.    cout << "{";
11.
12.    for (int i = 0; i < 10; i++)
13.    {
14.        if (g_used[i] == 1)
15.        {
16.            if (flag)
17.            {
18.                cout << ", ";
19.            }
20.            cout << set[i];
21.            flag = true;
22.        }
23.    }
24.
25.    cout << "}" << endl;
26. }

```

```

27.
28.
29. template<class T>
30. void SubSet(T set[], int k, int m)
31. {
32.     if (k > m)
33.     {
34.         return Output(set);
35.     }
36.
37.     g_used[k] = 0;
38.     SubSet(set, k + 1, m);
39.     g_used[k] = 1;
40.     SubSet(set, k + 1, m);
41. }
42.
43. int main(int argc, char *argv[])
44. {
45.     char str[10];
46.     cout << "请输入一个字符数组:";
47.     cin >> str;
48.     SubSet(str, 0, strlen(str) - 1);
49.     system("PAUSE");
50.     return EXIT_SUCCESS;
51. }

```

 **blog.csdn.net**

 <http://blog.csdn.net/ligt0610/article/details/7346175>

 <http://goo.gl/ffgS>

QR Code

