

```

1  /*
2  给定一组不含重复元素的整数数组 nums，返回该数组所有可能的子集（幂集）。
3
4  说明：解集不能包含重复的子集。
5
6  示例：
7
8  输入：nums = [1,2,3]
9  输出：
10 [
11     [3],
12     [1],
13     [2],
14     [1,2,3],
15     [1,3],
16     [2,3],
17     [1,2],
18     []
19 ]
20
21 来源：力扣（LeetCode）
22 链接：https://leetcode-cn.com/problems/subsets
23 著作权归领扣网络所有。商业转载请联系官方授权，非商业转载请注明出处。
24 */

```

分析:

- 根据二项式展开公式,若`nums`的长度为 n ,则子集合的个数共有 2^n .我们可以用二进制的1或0表示在子集中是否包含.
 - 缺点: `unsignedint`只能表示32个二进制位,只能表示 $n \leq 32$ 的数组长度的全集.

方法一:C++_二进制位遍历

```

1  class Solution
2  {
3      public:
4          vector<vector<int>> subsets(vector<int>& nums)
5          {
6              vector<vector<int>> ret_val;
7              vector<int> temp;
8              unsigned long int i = 0;
9              unsigned char j = 0;
10             unsigned long int size = nums.size();
11             unsigned long int size_pow = pow(2, size);
12
13             for (i = 0; i < size_pow; i++)
14             {
15                 temp.clear();
16                 for (j = 0; j < size; j++)
17                 {
18                     if (i & (0x01 << j))

```

```
19         {
20             temp.push_back(nums[j]);
21         }
22     }
23     ret_val.push_back(temp);
24 }
25 return ret_val;
26 }
27 };
28
29 /*
30 执行结果:
31 通过
32 显示详情
33 执行用时 :8 ms, 在所有 C++ 提交中击败了91.24% 的用户
34 内存消耗 :9 MB, 在所有 C++ 提交中击败了84.04%的用户
35 */
```

AlimyBreak
2019.10.03