

Given a set of distinct integers, nums return all possible subsets (the power set)

ACKTRACKING >> begin with empty set NUMS = [1,2,3] include in nums, you either add it to the set or don't [3] [2] [2,3] [1] [1,3] [1,2] [1,2,3] when you have recursively repeated for all the elements, you will have all possible subsets

## The CODE Dose

riterate through

```
def subsets_util(nums, i, s, powerSet):
                             add set to power Set
    if i >= len(nums):
        powerSet.append(s)
                               if reached end of
    else:
Include temp = s.copy()
                               nums
     temp.append(nums[i])
        subsets_util(nums, i + 1, temp, powerSet)
        subsets_util(nums, i + 1, s, powerSet)
                               5 not include
               move to 2
class Solution: next element
    def subsets(self, nums: List[int]) -> List[List[int]]:
        powerSet = []
        if len(nums) == 0:
            return
        subsets_util(nums, 0, [], powerSet)
        return powerSet
      recursive
                               begin with empty
       function
```