117 . Subset Sum Problem

AIM: To Solve the Subset sum problem by using backtracking

PROGRAM:

```
def subset_sum_backtracking(nums, target):
    """ Function to solve Subset Sum Problem using backtracking """
    subset = []
    result = []
    nums.sort()
    def backtrack(start, target):
        if target == 0:
            result.append(subset[:])
            return
        for i in range(start, len(nums)):
            if nums[i] > target:
                break
            subset.append(nums[i])
            backtrack(i + 1, target - nums[i])
            subset.pop()
    backtrack(0, target)
    return result
nums = [2, 3, 7, 8, 10]
target = 11
print(f"Subset sums that add up to {target}:")
print(subset_sum_backtracking(nums, target))
```

OUTPUT:

```
Subset sums that add up to 11:
[[3, 8]]
```

TIME COMPLEXITY: $0(2^N)$