```java
class Solution {
    public void f(int i, int[] nums, List<List<Integer>> res, List<Integer> ds, int k, int sum) {
        if (i == nums.length) {
            if (sum == k) {
                res.add(new ArrayList<>(ds));
            }
            return;
        }

        if (sum + nums[i] <= k) {
            ds.add(nums[i]);
            f(i, nums, res, ds, k, sum + nums[i]);
            ds.remove(ds.size() - 1);
        }
        f(i + 1, nums, res, ds, k, sum);
    }

    public List<List<Integer>> combinationSum(int[] nums, int k) {
        List<List<Integer>> res = new ArrayList<>();
        f(0, nums, res, new ArrayList<>(), k, 0);
        return res;
    }
}
```

```java
class Solution {
    public void f(int i, int[] nums, List<List<Integer>> res, List<Integer> ds, int k, int sum) {
        if (sum == k) {
            res.add(new ArrayList<>(ds));
            return;
        }

        for (int j = i; j < nums.length; j++) {
            if (sum + nums[j] <= k) {
                ds.add(nums[j]);
                f(j, nums, res, ds, k, sum + nums[j]);
                ds.remove(ds.size() - 1);
            }
        }
    }

    public List<List<Integer>> combinationSum(int[] nums, int k) {
        List<List<Integer>> res = new ArrayList<>();
        f(0, nums, res, new ArrayList<>(), k, 0);
        return res;
    }
}
```