Name: Shuo Wu Student ID:452666 wustl key: wushuo

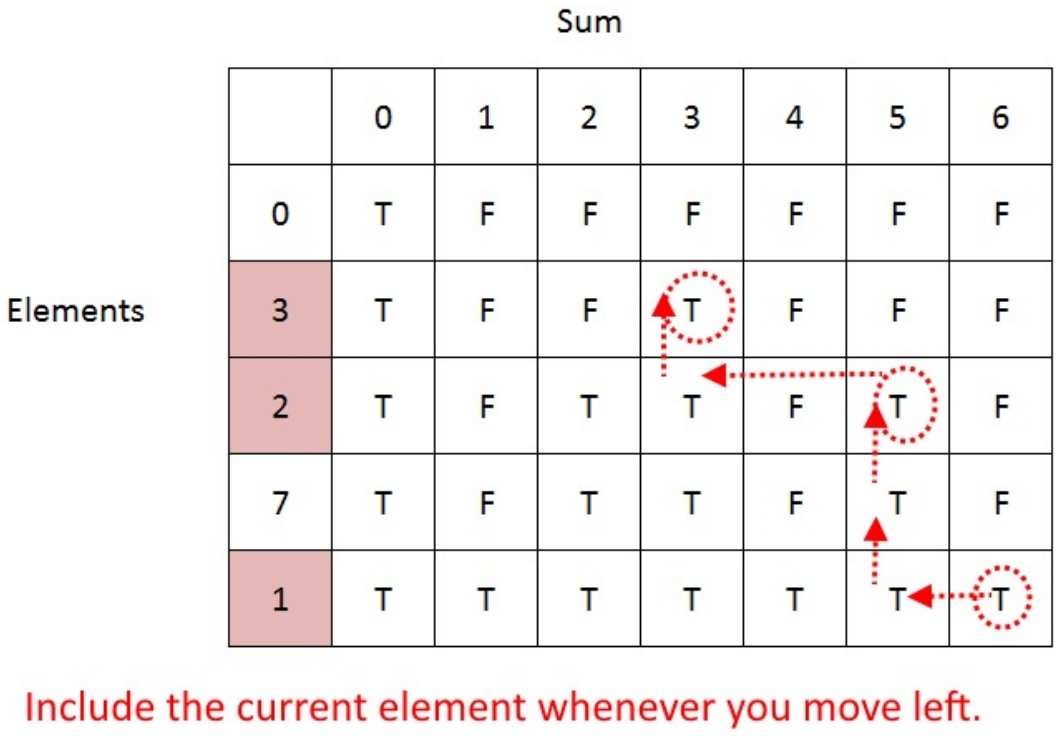
Q4,

Pseudo-code:
Step 1: check if there is a subsequence of the given sequence with sum equal to given sum:

```
def isSubSum(st, n, sm):
    # The value of subset[i][j] will be
    # true if there is a subset of
    # set[0..j-1] with sum equal to i

    boolean[][] solution = new boolean[A.length + 1][sum + 1]


    # If sum is 0, then answer is true
    for i in range(0, n + 1):
        subset[i][0] = True

    # If sum is not 0 and set is empty,
    # then answer is false
    for i in range(1, sm + 1):
        subset[0][i] = False

    # Fill the subset table in botton
    # up manner
    for i in range(1, n + 1):
        for j in range(1, sm + 1):
            if (j < st[i - 1]):
                subset[i][j] = subset[i - 1][j]
            if (j >= st[i - 1] and subset[i][j] == False):
                subset[i][j] = subset[i - 1][j] or subset[i - 1][j - st[i - 1]]

    return subset[n][sm];
```

		Sum						
		0	1	2	3	4	5	6
	0	T	F	F	F	F	F	F
Elements	3	T	F	F	T	F	F	F
	2	T	F	T	T	F	T	F
	7	T	F	T	T	F	T	F
	1	T	T	T	T	T	T	T

Step 2: if subset == true, return the sequence; else, return false
So I just include the current element whenever I move left:



Final code:

```
# Returns true if there is a subset
# of set[] with sun equal to given sum
def isSubsetSum(st, n, sm):
    # The value of subset[i][j] will be
    # true if there is a subset of
    # set[0..j-1] with sum equal to i
    subset = [[False for i in range(sm+1)] for y in range(n+1)]

    # If sum is 0, then answer is true
    for i in range(0, n + 1):
        subset[i][0] = True

    # If sum is not 0 and set is empty,
    # then answer is false
    for i in range(1, sm + 1):
        subset[0][i] = False

    # Fill the subset table in botton
    # up manner
    for i in range(1, n + 1):
        for j in range(1, sm + 1):
            if (j < st[i - 1]):
                subset[i][j] = subset[i - 1][j]
            if (j >= st[i - 1]):
                subset[i][j] = subset[i - 1][j] or subset[i - 1][j - st[i - 1]]

    print(subset[n][sm])
    FindSequence(subset,st,n,sum)
```

Name: Shuo Wu Student ID:452666 wustl key: wushuo

```
def FindSequence(subset,st,n,sum):
    answer = []
    row = n
    col = sum
    if subset[row][col] == False:
        return False
    else:
        while row>=1 and col >=1:
            if subset[row][col] == True:
                if subset[row-1][col] == False:
                    col = col -1
                    if st[row-1] not in answer:
                        answer.append(st[row-1])
                        sum = sum -st[row-1]
                        if (sum == 0):
                            break
                    continue
                else:
                    row = row -1
                    continue
            else:
                col = col -1
    #for i in range(0,len(answer)):
    print(answer)
    #print(subset)
    return 0


# Driver program to test above function
set = [2,7,1,3]
sum = 6
n = len(set)
print(isSubsetSum(set, n, sum))
```

output:

input: set = [3,2,7,1] sum =6
output answer = [1,2,3]
worst case: $T(n) = \Theta(sum*n)+ \Theta(n+sum)= \Theta(sum*n)$ since we need to fill the table with size sum*n.
where n is the length of the sequence. Sum is the number we want to find.