Yuxfan Li (Voldy)
CS325-001

Practice Assignment 2

1.

(a) i: End with i'th element in array
k: The sum of some subset is k
T[i,k]: The true or false of the sum of some subset that in array which end with i'th element. If there is some subset have a sum that equal to k, T[i,k]=1, If not, T[i,k]=0

Dynamic Programming Table.

i \ k	0	1	2	...	K
0	1	0	0	0	0
1	1				
2	1				
⋮	1				
n	1				

(b)
$$T[i,k] = \begin{cases} 1 & \text{if } T[i-1,k]==1 \text{ or } a_i == k \\ T[i-1, k-a_i] & \text{if } a_i < k \\ 0 & \text{if } a_i > k \text{ and } T[i-1,k] != 1 \end{cases}$$

(c) First, as the formula's meaning, T[0, 0→k]=1, T[0→n, 0]=0
Second, ∀ pair(i,k), we check if T[i-1, k]==1 and then if no check a_i if $a_i == k$, if not, check if $a_i > k$, if it is, T[i,k]=0, if no check T[i-1, k-a_i], and T[i,k] = T[i-1, k-a_i]

(d)

```
for i = 0 to n
    T[i,0] ← 0
for k = 0 to K
    T[0,k] ← 1
for i = 1 to n
    for k = 1 to K
        if T[i-1,k] == 1
            T[i,k] = 1
        else if ai == k
            T[i,k] = 1
        else if ai > k
            T[i,k] = 0
        else
            T[i,k] = T[i-1, k-ai].
```

(e) The first for-loop is n
The second for-loop is K
The third for-loop is $n \cdot K$

∴ Total running time is $\Theta(n+K+n \cdot K) = \Theta(n \cdot K)$

Yunfan Li (Voldy)
CS325-001

2.

(a) i : The location in QVH
P[i] : The max profit when stand at i'th location.

Dynamic Programming Table

i	0	1	2	3	...	n
P[i]						

(b) $$P[i] = \max \begin{cases} \max\{P[j] + \begin{cases} 1 & m_i - m_j \geq k \\ 0 & m_i - m_j < k \end{cases} \} * p[i] \\ p[i] \end{cases}$$

(c) First, P[0] = 0, for location 1 to n, use the formula shown above to calculate the P[i]

(d)
```
P[0] ← 0
max_location ← 0
for i = 1 to n
    if m_i − m_maxlocation >= k
        flag = 1
    else
        flag = 0
    temp = P[max_location] + flag * p[i]
    if temp > p[i]
        P[i] = temp
    else
        P[i] = p[i]
    if P[i] > P[max_location]
        max_location = i;
```

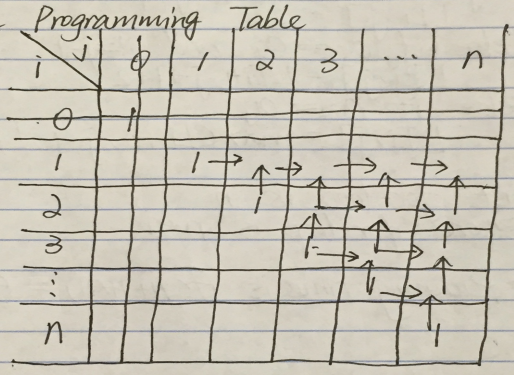
(e) The for-loop running time is n
The total running time of this algorithm is $\Theta(n)$

Yuntan Li (Voldy)
CS325-001

3.

(a) i : The string start with ith character
j : The string end with jth character
L[i,j] : The ~~len~~ max length of palindromic subsequence in string i to j.

Dynamic Programming Table



(b) $L[i,j] = \begin{cases} \max\{L[i+1,j], L[i,j-1]\} & \text{if } a_i \neq a_j \\ L[i+1,j-1]+2 & \text{if } a_i == a_j \text{ and } i \neq j-1 \\ 2 & \text{if } a_i == a_j \text{ and } i = j-1 \end{cases}$

(c) First, the base case is for all L[i,j] such that i==j, L[i,j] = 1
~~Second, ∀pairs (i,j), check that if $a_i == a_j$, if it is, then L[i,j] = L[i+1,j-1]; if not, find the max between L[i+1,j] and L[i,j-1]~~
Second, ~~also~~ for the first time, j = i+1, ∀pairs (i,j), check if $a_i == a_j$ and i = j-1, the L[i,j] = 2, if i ≠ j-1 and a_i still equal to a_j, then L[i,j] = L[i+1,j-1]+2; if $a_i \neq a_j$, then L[i,j] = max of L[i+1,j] and L[i,j-1]. For the kth time, j = i+k, do the process above.

(d)
```
for i = 1 to n
    L[i, i] = 1
for k = 1 to n
    for i = 1 to n
        j = i + k
        if a_i == a_j and i = j - 1
            L[i, j] = 2
        else if a_i == a_j and i != j - 1
            L[i, j] = L[i+1, j-1] + 2
        else if a_i != a_j
            L[i, j] = max(L[i+1, j], L[i, j-1])
```

(e) The first for-loop is n
The second for-loop is $n*n$

∴ Running time is $\theta(n+n*n) = \theta(n^2)$