Practice Assignment 2

Andrew Johnson

Psuedo

1)
```
a = [a_1, ..., a_n]
Positive integer K
bool memoSol[][] = [n][K]
for s = [0, ..., K]
    memoSol[0,s] = False
for i = [0, ..., K]
    memoSol[i,0] = True
return hasSubset(n, K)

hasSubset(j, total)
    if memoSol[j][total] != null
        return memoSol[j][total]
    if j == 1
        memoSol[j][total] = (a_j == total)
        return a_j == total
    result = hasSubset(i-1, total-a_j) || hasSubset(i-1, total)
    memoSol[j][total] = result
    return result
```

DP Table

A

s \ i	0				n
0	T	T	T	T	T
	F				
	F				
K	F				Start

Start at A[n][K], recursively call the block to the left or a block to the left and minus a_i from K. If it hits K=0 before or when n=0, return True all the way back up, if not return false.

This Running time is $O(nK)$

$$P[i] = \max \begin{cases} \max_{j<i} \{ P[j] + f(m_i, m_j) \cdot P_i \} \\ P_i \end{cases}$$

$$f(m_i, m_j) = \begin{cases} 0 & \text{if } m_i - m_j < K \\ 1 & \text{if } m_i - m_j \geq K \end{cases}$$

2) $p_i = [P_1, P_2, \ldots, P_n]$

$P(1 \ldots n)$
prev$(1 \ldots n)$

To fill the DP Table it starts at (0,0) and grows out.

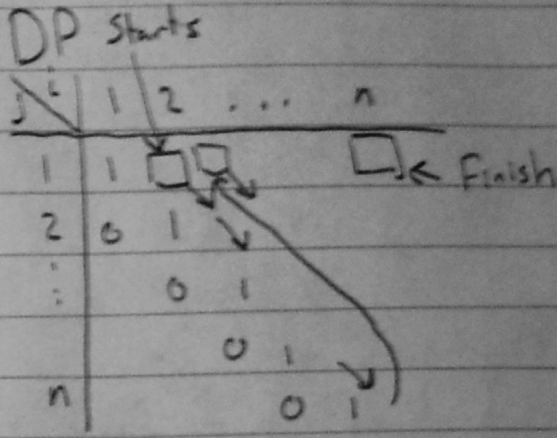
```
prev(1) = 0
P(1) = 0
for i = [2,...,n]
    prev(i) = prev(i-1)
    if (m_i - m_{i-1} >= K)
        prev(i) = i-1

for i = [2,...,n]
    P(i) = max{p_i + P(prev(i)), P(i-1)}

return P(n)
```

This only takes $O(n)$

3)
```
n = length of sequence
for i = 1, ..., n
    P[i][i] = 1
    for j = 1, ..., i-1
        P[i, j] = 0
for j = 2, ..., n
    for i = j-1, ..., 0
        P[i, j] = P[i, j-1]
        if P[i+1, j-1] ≥ P[i, j-1]
            P[i, j] = P[i+1, j]
        if x[i] == x[j]
            P[i, j] = P[i+1, j-1] + 2
return P[1, n]
```



This has Runtime of $O(n^2)$

Filling DP table, fill diagonal with 1's and all cells below with 0, then start at P[1,2] and starting and filling in diagonally to the right until you hit P[1, n] which is your answer.