Homework 3

Wednesday, February 27, 2019 5:30 PM

1.) n:10 d(1): 1 d(2): 5 d(3):8

C	Value	Collection
0	0	< >
1	1	< 1 >
2	2	< 1, 1 >
3	3	< 1, 1, 1 >
4	4	< 1, 1, 1, 1 >
5	1	< 5 >
6	2	< 5, 1 >
7	3	< 5, 1, 1 >
8	1	< 8 >
9	2	< 8, 1 >
10	2	< 5, 5 >

$\langle d_1, \ldots, d_k \rangle$

Let C[i] be the fewest # of coins for changing pennies.

$$C[i] = \begin{cases} \min_{1 \le j \le k,\ d_j \le i} \{ C[i-d_j] \} + 1 & i > 0 \\ 0 & i = 0 \end{cases}$$

$C[i] = \min \{ C[i-d_j] \} + 1$

1.) $C[1] = \min \{ C[1-1] \} + 1 = \min \{ C[0] \} + 1 = \langle 1 \rangle$

2.) $C[2] = \min \{ C[2-1] \} + 1 = \min \{ C[1] \} + 1 = \langle 1, 1 \rangle$

3.) $C[3] = \min \{ C[3-1] \} + 1 = \min \{ C[2] \} + 1 = \langle 1, 1, 1 \rangle$

4.) $C[4] = \min \{ C[4-1] \} + 1 = \min \{ C[3] \} + 1 = \langle 1, 1, 1, 1 \rangle$

5.) $C[5] = \min \{ C[5-1], C[5-5] \} + 1 = C[0] + 1 = \langle 5 \rangle$

6.) $C[6] = \min \{ C[6-1], C[6-5] \} + 1 = C[1] + 1 = \langle 5, 1 \rangle$

7.) $C[7] = \min \{ C[7-1], C[7-5] \} + 1 = C[2] + 1 = \langle 5, 1, 1 \rangle$

8.) $C[8] = \min \{ C[8-1], C[8-5], C[8-8] \} + 1 = C[0] + 1 = \langle 8 \rangle$

9.) $C[9] = \min \{ C[9-1], C[9-5], C[9-8] \} + 1 = C[1] + 1 = \langle 8, 1 \rangle$

10.) $C[10] = \min \{ C[10-1], C[10-5], C[10-8] \} + 1 = C[5] + 1 = \langle 5, 5 \rangle$

2.) A.)

$$C[i,j] = \begin{cases} C[i-1, j-1] + C[i-1, j] & i \ge j \text{ and } j \ge 1 \\ 1 & j = 0 \text{ or } j = i \end{cases}$$

B.)

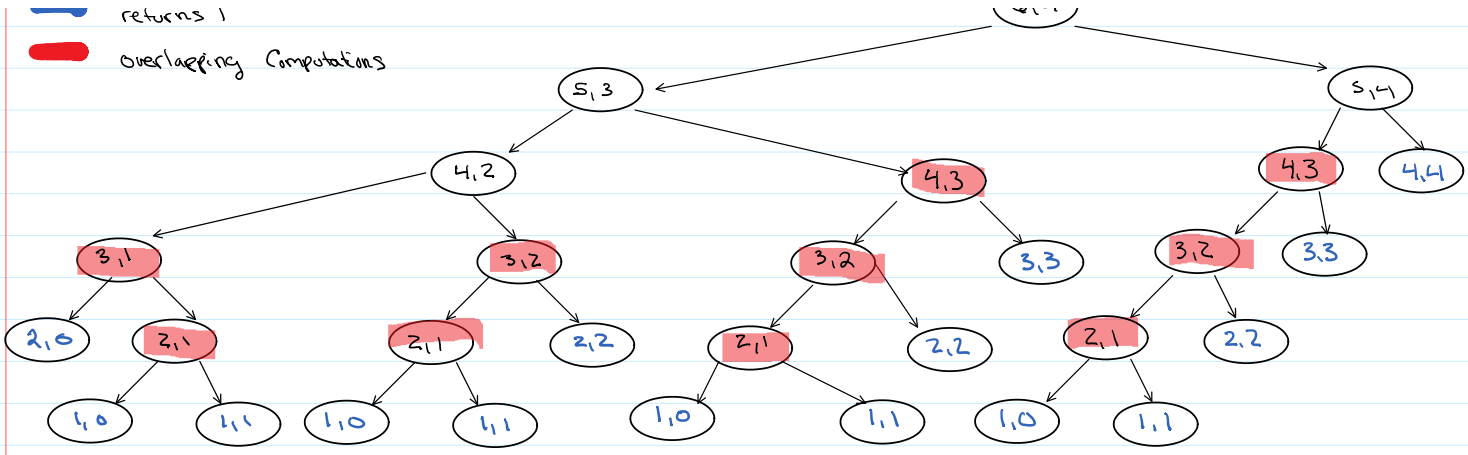
```
PascalsTriangle (i, j)
  if j = 0 or j = i :
     return 1
  return PascalsTriangle(i-1, j-1) + PascalsTriangle(i-1, j)
```

returns 1

Overlapping Computations

returns 1

Overlapping Computations



C.)

```
PascalsTriangle( numRows)                    O(n²)
  Array[numRows][numRows]
  for i:=0; i≤n; i++:                        O(n)
      for j=0; j≤i; j++:                     O(n)
        if i==j:
          Array[i][j]=1
        if j==0:
          Array[i][j]=1
        else:
          Array[i][j]=(Array[i-1][j-1] + Array[i-1][j])
```

3.) $X = \{A, C, T, C, C, T, G, A, T\}$

$Y = \{T, C, A, G, G, A, C, T\}$

$LCS(X, Y) = T, C, G, A, T$

$$c[i,j] = \begin{cases} \max(c[i,j-1], c[i-1,j]) & i \neq 0 \text{ and } j \neq 0 \text{ and } X_i \neq Y_j \\ c[i-1,j-1]+1 & i \neq 0 \text{ and } j = 0 \text{ and } X_i = Y_j \\ 0 & i = 0 \text{ or } j = 0 \end{cases}$$

	0	T	C	A	G	G	A	C	T
0	0	0	0	0	0	0	0	0	0
A	0	0↑	0↑	1↖	1←	1←	1↖	1←	1←
C	0	0↑	1↖	1↑	1↑	1↑	1↑	2↖	2←
T	0	1↖	1↑	1↑	1↑	1↑	1↑	1↑	3↖
C	0	1↑	2↖	2←	2←	2←	2←	2↖	3↑
C	0	1↑	2↖	2←	2↑	2↑	2↑	3↖	3↑
T	0	1↑	2↑	2↑	2↑	2↑	2↑	3↑	4↖
G	0	1↑	2↑	2↑	3↖	3↖	3←	3↑	4↑
A	0	1↑	2↑	3↖	3↑	3↑	4↖	4←	4↑
T	0	1↑	2↑	3↑	3↑	3↑	4↑	4↑	5↖
		T	C			G	A		T

4.) $A = [10, 15, 22, 19, 33, 5, 50, 95]$

$A = [10, 15, 22, 19, 33, 5, 50, 95]$

$n = 6$

0 1 2 3 4 5 6 7

10 15 22 19 33 5 50 95

j j jj ij jj i i

SubSeq = [1, 2, 3, 1, 4, 1, 5 (6)]

```
LIS( Array A, int n):
    SubSeq: [n]
    for i:0 to n:
      SubSeq[i] = 1

    for i=1 to n:
       for j:0 to i:
            if A[i] > A[j]:
              if SubSeq[i] < SubSeq[j]+1:
                 SubSeq[i] = SubSeq[j]+1

    Max: 0
    for i=0 to n:
       Max: max < SubSeq[i]? SubSeq[i] : max
    return Max
```

5.) S: [1, 2, 4, 10]

t = 11

A.)

		y_1	y_2	y_3	y_4	y_5							y_{12}
		0	1	2	3	4	5	6	7	8	9	10	11
X_1	1	T	T	F	F	F	F	F	F	F	F	F	F
X_2	2	T	T	T	T	F	F	F	F	F	F	F	F
X_3	4	T	T	T	T	T	T	T	T	F	F	F	F
X_4	10	T	T	T	T	T	T	T	T	F	F	T	T

There is a Subset

Rows = X

Column = Y

T = you can get Y by some combination of X values up to the current row's X value

F = Otherwise

The last Row/Column states if there is a subset whos values equal t

B.) if the X value is greater than the Y value take the value from the previous row's Y value

if the X value is not greater than the Y value and the Xvalue - Yvalue > 0 take the previous rows Y value - (Xvalue - Yvalue) entry

C.)

```
Subset( Set S, int target)
    n = S.size
bool T = [n+1][target+1]

    for i = to n
        T[i][0] = 0

    for i = 1 to n:
        for j=1 to target:
            if( j - S[i-1] ≥ 0):
                T[i][j] = T[i-1][j] OR T[i-1][j-S[i-1]]
            else:
                T[i][j] = T[i-1][j]

    return T[n][target]
```