Dynamic Programming

Coin Change problem

Coins = {1, 5, 10, 25, 100}.

You must make 273 in the fewest number of coins:

What would a greedy algorithm do?

Is greed always correct?

coins = {1, 5, 6}

make 10

1, 5 11 20 50

What would Dynamic Programming do?

Coin denoms are $d_1, d_2, d_3, \ldots, d_n$

$$\text{LeastCoins}(K,i) = \begin{cases} 0 & \text{if } K = 0 \\ \infty & \text{if } K < 0 \\ \min\begin{pmatrix} 1 + \text{LeastCoins}(K-d_i, i) \\ \text{LeastCoins}(K, i-1) \end{pmatrix} \end{cases}$$

d_i is max denom that we can use in making Change $= K$

$d_1 = 1 \quad d_2 = 5 \quad d_3 = 6 \quad \text{Amount} = 10$

We are interested in LeastCoins(Amount, n)

K =	0	1	2	3	4	5	6	7	8	9	10
Least # coins	0	1	2	3	4	1	1	2	3	4	2
last coin	1	1	1	1	1	5	6	1	1	1	5

Dynamic Programming - Dance Contest!

Input:

	1 Cha-cha	2 Waltz	3 Western Swing	4 Lindy Hop	5	6	7	8
Points	15	5	3	10	4	6	11	18
Skip	5	1	2	3	2	1	2	3
dance	15+24 = 39	5+28 = 33	3+24 = 27	10+18 = 28	4+18 = 22	6+18 = 24	11	18
don't dance	33	28	28	24	24	18	18	0

max

accrue in each one ... and how much rest you will need (in which case you will need to **Skip** a certain number of dances

We take skip[i] = 1 to mean we can go on to the i+1 dance. (if it exists)

$$Dance[i] = Points[i] + \max\Big(Dance[i + Skip[i]],\ Don't\ Dance[i + Skip[i]]\Big)$$

$$Don't\ Dance[i] = \boxed{}$$

for you to do