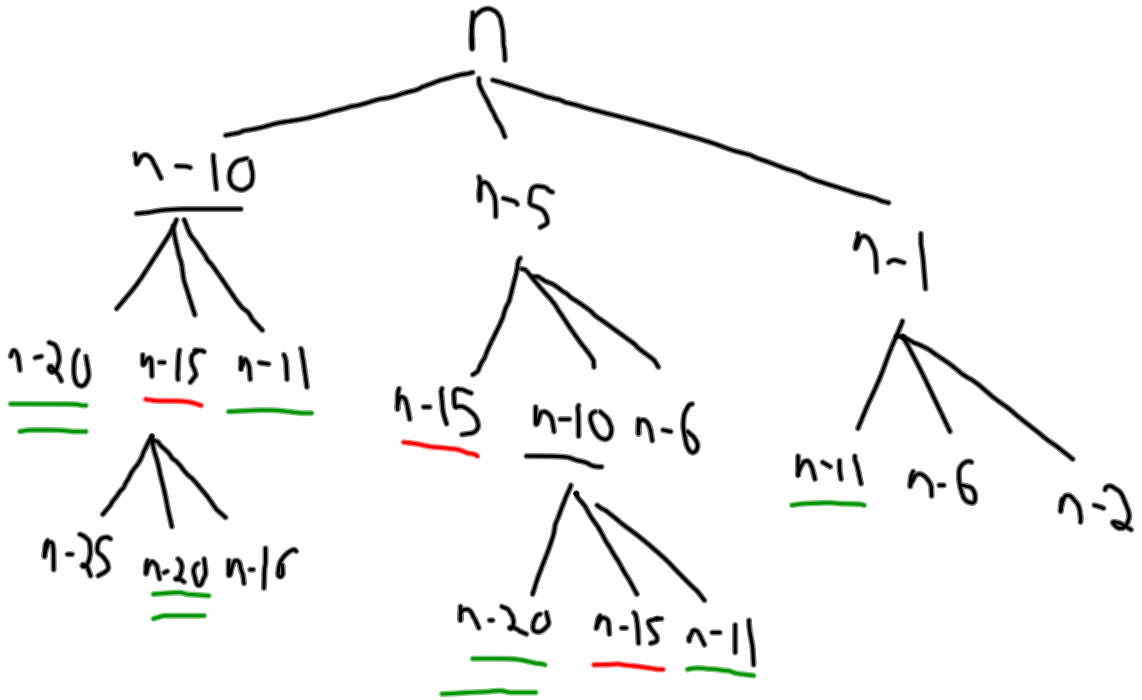


Change for 67 cents.

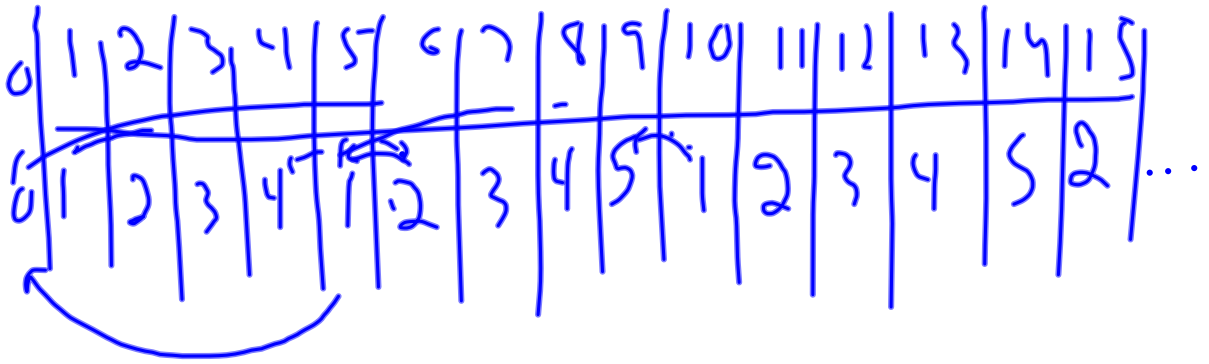
Figure out how to make change for every value 1..66 cents

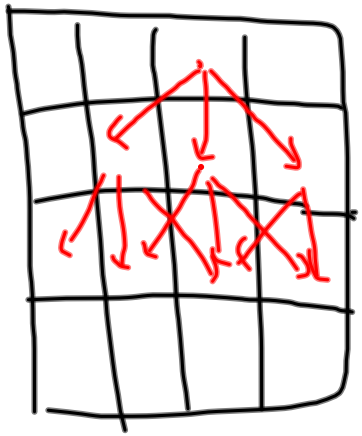
Tree has $> \frac{n}{10}$ levels. Branches by factor of 3

$> 3^{n/10}$ time.

- Recognize that there are many repeated subproblems, but only p unique subproblems.

- If I compute them bottom-up $(1..p)$, I don't need recursion.





4x4 grid

n x n grid

Start at top, move down straight or diag.

How many paths from top to bottom?

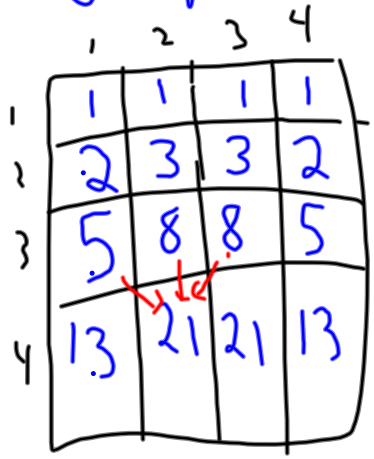
(King on row 1, how many forward paths to row n.

Upper bound: $\leq n\,3^{n-1}$.

Compute an exact bound

Could enumerate paths (exponential)

Instead. Compute # of paths from top to every square (in bottom up fashion)



$C(x)$ = # paths to square x

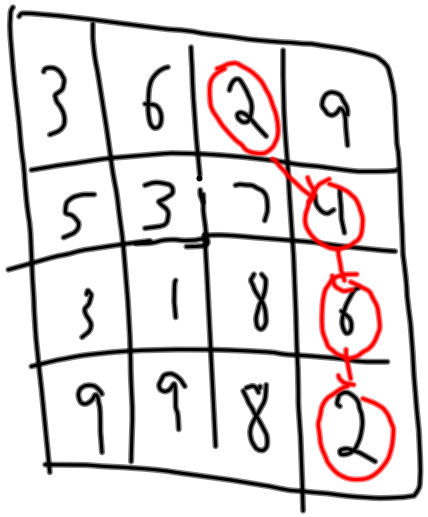
$O(n^2)$ time alg.

$e(i,j)$ = # of paths to (i,j)

$$e(i,j) = e(i-1,j-1) + e(i-1,j) + e(i-1,j+1)$$

modulo boundary conditions

Suppose each square has a cost c_{ij}



pay c_{ij} when travelling through (i,j)

Cheapest top-to-bottom path

Solution is similar to the one for counting paths