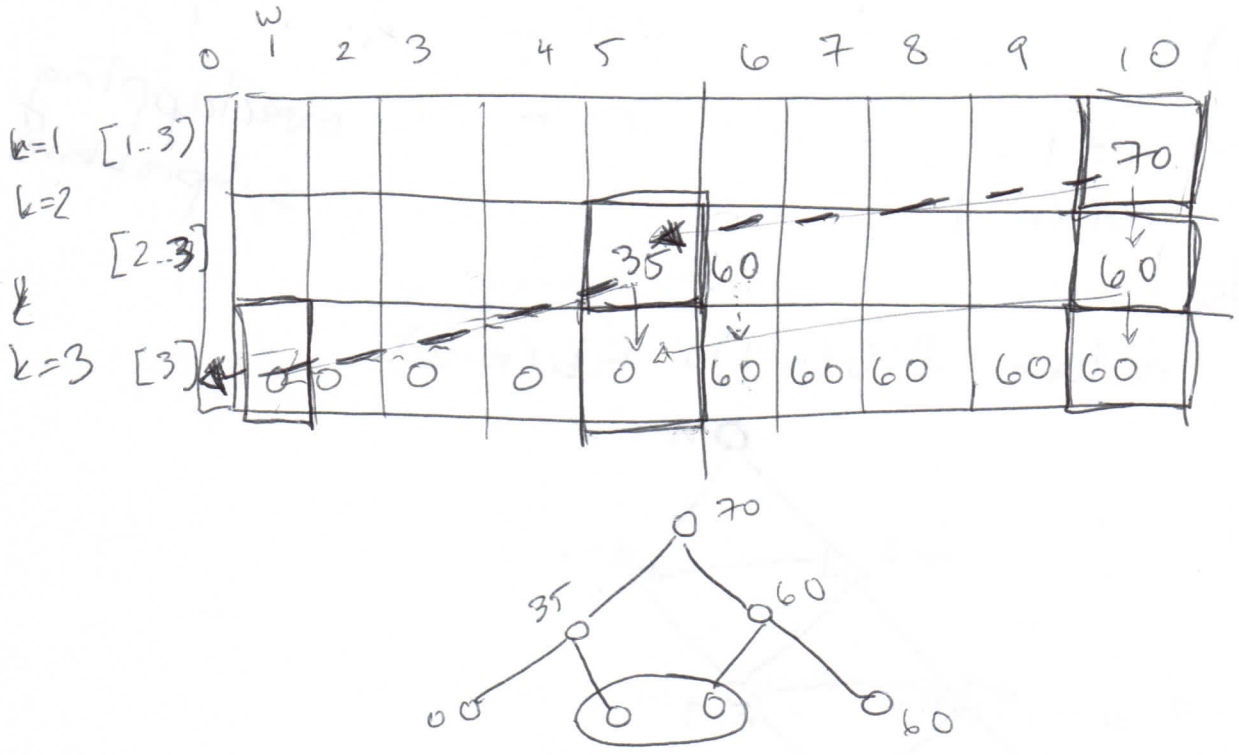
[(35, 5) (35, 5) (60, 6)] W = 10



k levels of recursion ⇝ 2^k recursive calls

As soon as $2^k \geq W$, must be two repeated recursive calls

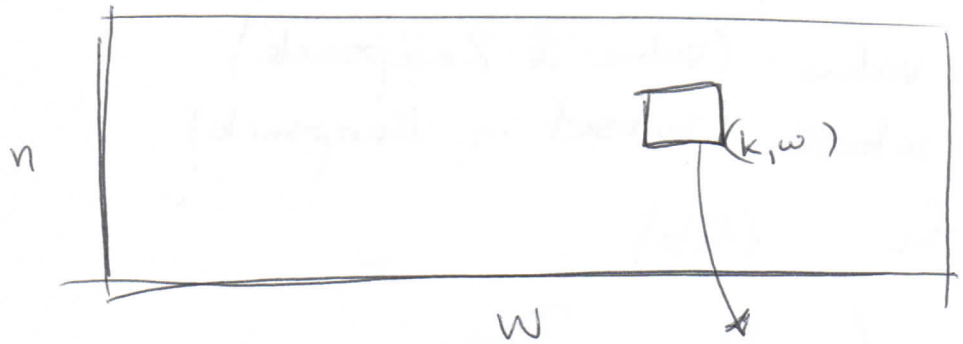
If $2^n \gg W$, then Dyn. Prog. better

But can be better anyways.

How to find solution (not just its value).

One way: Maintain an auxiliary table.

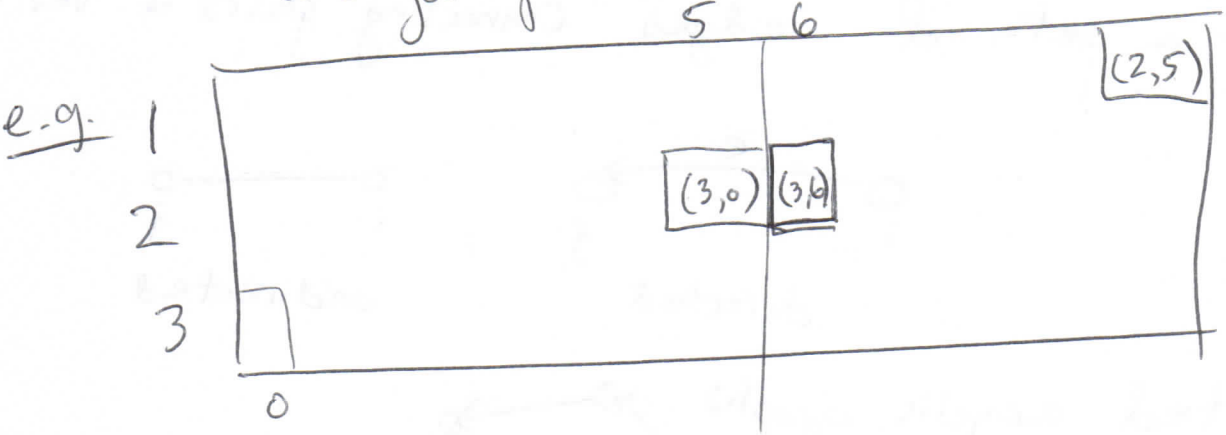
Same size (Knapsack: $n \times W$)



lst[k..n] w/ threshold w

(k,w) entry is (k',w'): (k',w') is the one you picked from the recurrence relation

e.g.



Innermost conditional

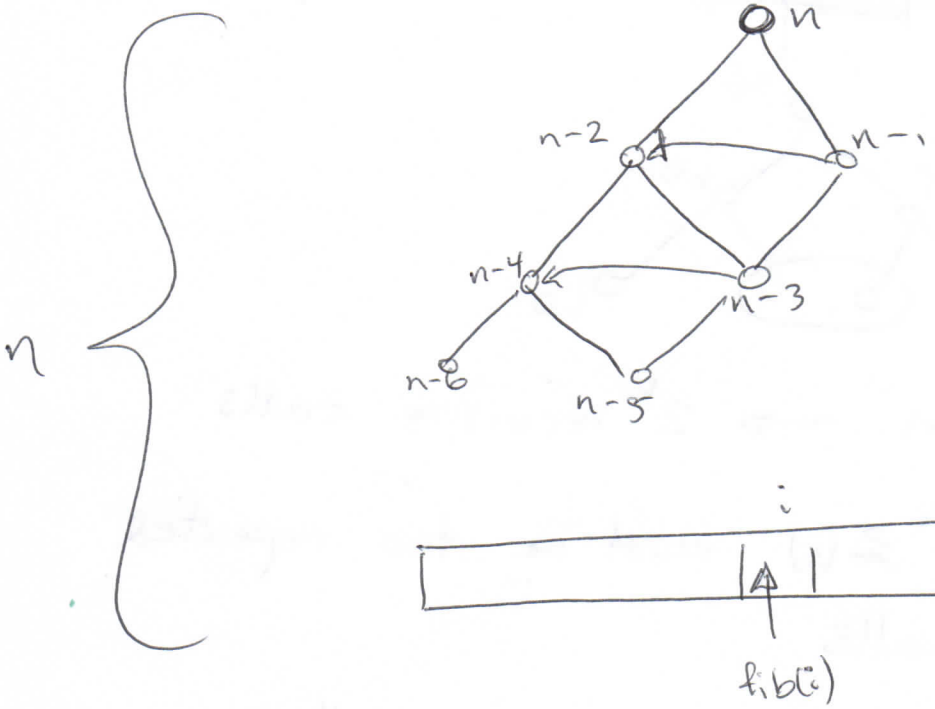
```
if wt_i > w
    table(i,w) = table(i+1, w)
    aux(i,w) = (i+1, w)

else
    val_taken = val_i + table(i+1, w-wt_i)
    val_not = table(i+1, w)
    if val_taken ≥ val_not:
        table(i,w) = val_taken
        aux(i,w) = (i+1, w-wt_i)
    else
        table(i,w) = val_not
        aux(i,w) = (i+1, w)
```

$F_n = F_{n-1} + F_{n-2}$ $\quad F_0 = F_1 = 1$

```
fib(n) =
    if n ≤ 1:
        return 1
    else
        return fib(n-1) + fib(n-2)
```

overlapping subproblems



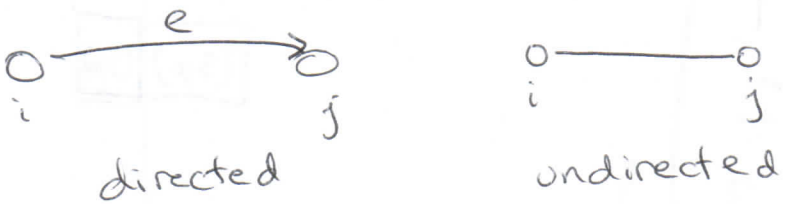
template


```
better-fib(n):
    arr of length n
    arr[0] = arr[1] = 1
    for i = 2, ..., n
        arr[i] = arr[i-1] + arr[i-2]
    return arr[n]
```

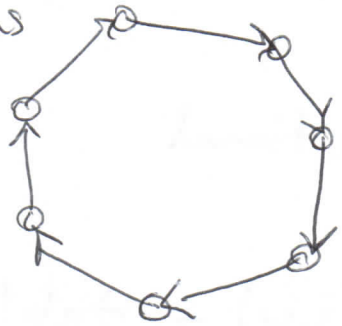
Optimum value (value of Knapsack)
Optimum selection (subset in Knapsack)
computing fn (fib)

Counting w/ Dyn. Prog.

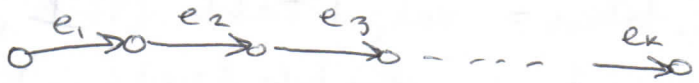
A graph has a set of vertices (nodes)
+ a set of edges connecting pairs of verts



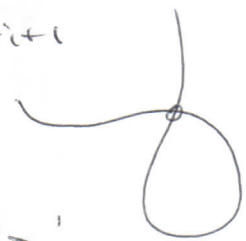
Directed acyclic graphs
i.e. no cycles.



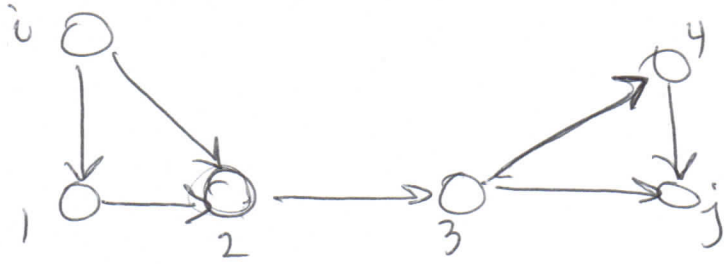
A path in a directed graph is a sequence of edges $(e_1, \dots, e_k)$



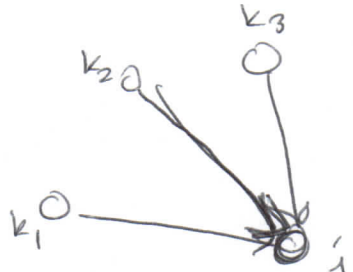
endpoint of e_i = start point of e_{i+1}



Given: A DAG, two vertices i, j
Goal: Compute the # paths from $i \to j$



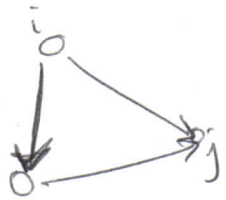
A DAG.



all verts k
w/ edges $k \to j$

$$\#\text{ paths } i \to \boxed{j} = \sum_{\substack{k: \\ \text{edge } k \to j}} \#\text{ paths } i \to \boxed{k}.$$

Base case: if there's an edge $i \to j$
paths of length 1 is 1.



i	1	2	3	4	j
	1	(2)	2	2	4

2

i 2 j