Algorithm

Fibonacci

$Fib(n) = Fib(n-1) + Fib(n-2)$

Rod Cutting Problem

Total Possible Cut = 2^{n-1}

$q = \max(q, P[i] + \text{CUT-ROD}(P, n-i))$

Matrix Chain Multiplication

$c_{ij} = c_{ij} + a_{ik} \cdot b_{kj}$

$m[i,j] = m[i,k] + m[k+1,j] + P_{i-1} P_k \cdot P_j$

knapsack-0-1

$$B(k,w) = \begin{cases} B(k-1,w) & ; w_k > w \\ \max\{B(k-1,w),\ B(k-1, w-w_k) + b_k\} & ; \text{else} \end{cases}$$

BFS

Uses Queue

DFS

Stack on Recursive Call

Strongly Connected

Run DFS(G)

Run DFS(G^T)

MST

Find the ~~light ed~~ safe edge

KRUSKAL

MAKE - SET(v)

Sort

Find Set

Prims Algorithm

Find a light Edge

min-Heap based on key

Dijkstra (G,w,s)

Single Source

No negative edge

- initialize
  - extract min
  - ~~Bell~~
  - Relax with adjacent

Bellman

identify negative edge

- single source

returns True or False

depend on no negative cycle

- RELAX all edge

Check again one more

DAG Shortest Path

Topological sort

Relax according to the sort

~~Bellm~~ Floyed Warshell

$$d_{ij}^{(k)} = \min\left(d_{ij}^{(k-1)},\ d_{ik}^{(k-1)} + d_{kj}^{(k-1)}\right)$$

realme Shot by Legend T.JOY

Complexity

Fibonacci

Recursive = $O(1.6^n)$

Dynamic Both = $O(n)$

Space = $O(n+n)$
= $O(n)$

Rod Cutting Problem

Recursive = $O(2^n)$

Dynamic Both = $\Theta(n^2)$

space = $O(n)$

Matrix Chain Multiplication

$T(n) = O(n^3)$

~~$O(n^2)$ =~~

Space = $O(n^2)$

Knapsack - 0-1

Brute-Force = $O(2^n)$

Dynamic = $O(n * w)$

BFS

$O(V+E)$

DFS

$O(V+E)$

DFS-VISIT = $O(E)$
called $O(V)$ times

Topological Sort

$\Theta(V+E)$

because of DFS

Strongly Connected Component

$\Theta(V+E)$

~~DF~~
DFS = G^T = $\Theta(V+E)$

Kruskal (G,w)

$O(E \lg E)$

because of sort

$O(V + E \lg E + E \lg V)$

MST- PRIM (G,w,r)

$O(V) + O(V \lg V + E \lg V)$

$= O(E \lg V)$

Dijkstra

$O(V \lg V + E \lg V)$

$= O(E \lg V)$

Bellman

$O(VE)$

DAG Shortest

$\Theta(V+E)$

because of topological sort

All Pair Shortest

Dij ⇒ $O(VE \lg V)$

Bellman = $O(V^2 E) = O(V^4)$

Dynamic

Floyed-warshell = $O(V^3)$