1

DFS algorithm searches every nodes (V) and every edges (E), therefore: O(V+E)

If the Adjacently Matrix is used, then the result is O(V^2)

2

Dijkstra's Algorithm can find the shortest path from one source to all other vertices while Floyd's can find the shortest path between all pairs of nodes

"But Dijkstra's can do pretty much the same with some modifications, such as marking the nearest node as new source, and the previous source as visited." – the colleague said.

The time complexity of Dijkstra's algorithm is O(E log(V)). So if you want to calculate all the edges, it would be

VE log(V)

= (V(V*(V-1))/2) * log(V)

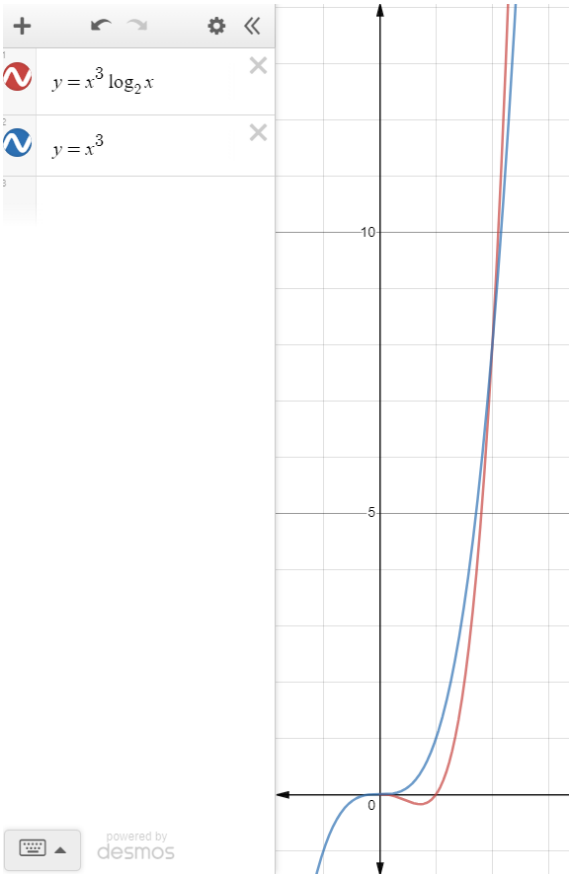
= V(V^2 - V)/2 * log(V)

= (V^3 – V^2) * log(V)

= V^3 * log(V) – V^2*log(V)

The complexity would be O(V^3 * log(V)) while Floyd's is just O(V^3), which is faster.

So just use Floyd's.



3

Depth-first search

A – B – F – I – N – E – J – K – O – C – L – D – G – M
\ H

Breadth-first search