**Shortest Path Algorithm:**

* Family of algorithms designed for solving shortest path problems.
* Shortest path problem => given 2 points A and B find shortest path between them.
* Two main types of shortest path algorithm: 1. single-source 2. All-pairs.
* Main types of algorithms: Bellman-Ford, Dijkstra, Topological sort, Floyd-Warshall, Johnson

Single-Source shortest path algorithm:

* Def’n: Given a graph G, with vertices V, edges E with weight function w(u,v) = Wu,v and a single source vertex s, return shortest paths from s to all other vertices in V.
* Bellman-Ford:
  + Solve single-source problem in general case, where edges can have negative weights and graph is directed. If graph is un-directed must make it directed by including two edges in each direction.
  + Has property that can detect negative weight cycles reachable from the source => no shortest path exist.
  + If no negative weight cycle, then Bellman-Ford returns weight of shortest path along with path itself.
  + Simpler than Dijkstra and well suited for distributed systems. But time complexity of O(VE).
  + Dynamic programming approach
* Dijkstra:
  + Uses Breadth First Search(Not a single source SPA) to solve single source problem.
  + Graph cannot have negative weight edges because of this Dijkstra improves runtime of Bellman-Ford.
  + Doesn’t work for graphs with negative weight edges -> why??
  + Greedy approach

All-pairs:

* Def’n: Given a graph G, with vertices V, edges E with weight function w(u,v) = Wu,v. Return shortest path from u to v for all (u,v) in V.
* Floyd-Warshall:
  + Use dynamic programming approach.
  + Can have negative weight edges.
  + Works well with dense graphs
* Johnson’s:
  + Works best for sparse graphs
  + Take advantage of the concept of reweighting and use Dijkstra’s algorithm on many vertices to find shortest path once it has finished reweighting the edges.

Applications:

* Maps ex: google map, kakao map, etc…
* Networks, operations, and logistic research

References:

* <https://brilliant.org/wiki/shortest-path-algorithms/>
* <https://medium.com/basecs/finding-the-shortest-path-with-a-little-help-from-dijkstra-613149fbdc8e>