

- longest paths in directed graphs
 - scoring of max score paths $\equiv$ alignments.
 - NP-complete problems intertracable $\equiv$ only brute force algs. avail. $\equiv$ exponential time algs.
- shortest path algs.
 - polynomial time algo.
 - shortest paths on edit-like graphs can be solved optimally in linear time.
 - edit-like graphs, DAG = directed acyclic graph

on DAG:

longest path

shortest path

}

in linear time

key concept: topological order

dijkstra's algo:

$G = \langle N, A \rangle$ directed graph
 N = set of nodes / vertices
 A = set of edges; each edge has a non-neg cost.
 $N = \{1, 2, \dots, n\}$
 1 = source node.

problem:
given: $G = (N, A)$
 $L = [i, j]$ = cost of edge (i, j)

compute: the cost of the shortest paths from the source to each of the other nodes in G .

- C, S two sets of nodes
- S = "special" nodes nodes already chosen
- C = the set of candidate nodes already chosen
- at any step S contains all the nodes whose minimal cost from the source node is already known (computed)
- C = the rest of the nodes
- at any step we choose a candidate node (where the cost to the source node is minimal cost) and add it to S .
- def: a path from the source to a node v is special if all intermedian nodes are in S .
- array D contains cost of shortest special path to each node in G .
- when we add a new v from C to S then the shortest special path to v is also the shortest path in the graph

$N = \{1, 2, \dots, n\}$
 $L[i, j] \geq 0$ if $A(i, j) \in A$
 $L[i, j] = \infty$ if $A(i, j) \notin A$

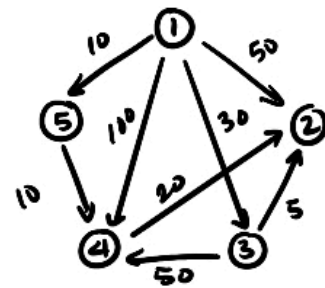
initialization:

1 = source node
 $S = \{1\}$
 $C = \{2, 3, \dots, n\}$
 $P[i] = 1$
 $i = 2, 3, \dots, n$
 $P[2, 3, \dots, n]$
 for $i = 2$ to n do
 $D[i] = L[1, i]$

greedy loop:

repeat $n-2$ times
 v = node $\in C$ with minimum value $D[v]$
 $S = S \cup \{v\}$
 $C = C \setminus \{v\}$
 for each $w \leftarrow C$ do
 if $D[w] > D[v] + L[v, w]$
 then
 $D[w] = D[v] + L[v, w]$
 $P[w] = v$

example:



step	v	C	D
initialization	-	$\{2, 3, 4, 5\}$	$[50, 30, 20, 10]$
1	5	$\{2, 3, 4\}$	$[50, 30, 20, 10]$
2	4	$\{2, 3\}$	$[40, 30, 20, 10]$
3	3	$\{2\}$	$[35, 30, 20, 10]$

topological sorting:

- a topological sort is on topological ordering of a directed graph is a linear ordering of the vertices such that for every directed edge from vertex u to vertex v , vertex u comes before v .
- a topological ordering is only possible if and only if the graph is a DAG, i.e. it has no directed cycles
- topo ordering if it exists can be found in linear time



$L = \{1, 2, 4, 3, 5, 7, 6, 8, 9\}$
 $S = \{1, 2, 4, 3, \dots\}$

$\xrightarrow{50} G$ shortest path
 $\xrightarrow{-50} -G$ longest path

topological sorting algorithm

for DAG $G = (V, E)$

L = the list of topological sorted vertices of V
 S = set of vertices with no incoming edges

while $S \neq \emptyset$ do
 remove node u from S
 add u to L at the local
 for each node v with edge $e = (u, v) \in E$
 remove edge e from graph
 if v has no incoming edges
 put v into S