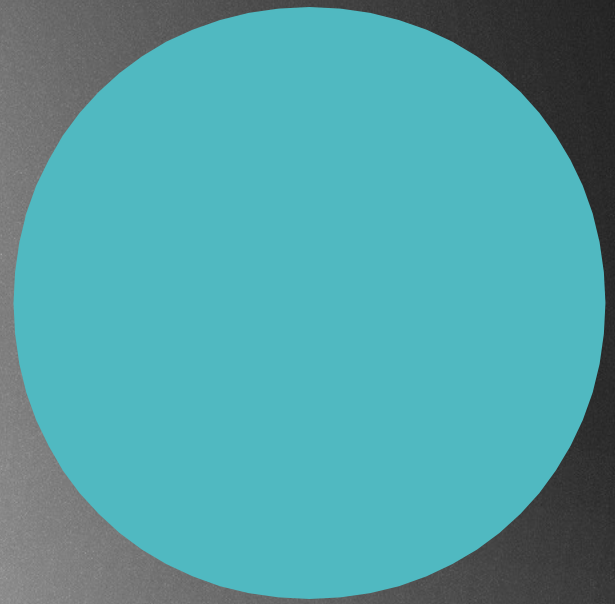
SHORTEST PATH

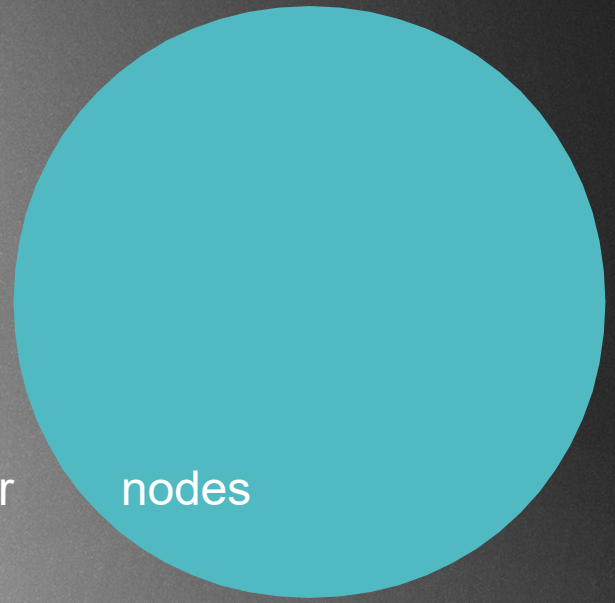


APPLICATIONS OF SHORTEST PATH ALGORITHMS

DAG shortest path

- If the graph is a DAG, so there is no directed cycles, it is easier to find the shortest path
- We sort the vertices into topological order: we iterate throught the topological order relaxing all edges from the actual vertex
- Topological sort algorithm computes shortest path tree in any edge weighted (can be negative!!!) DAG in time O(E+V)
- It is much faster than Bellman-Ford or Dijkstra
- Applications: solving Knapsack problem

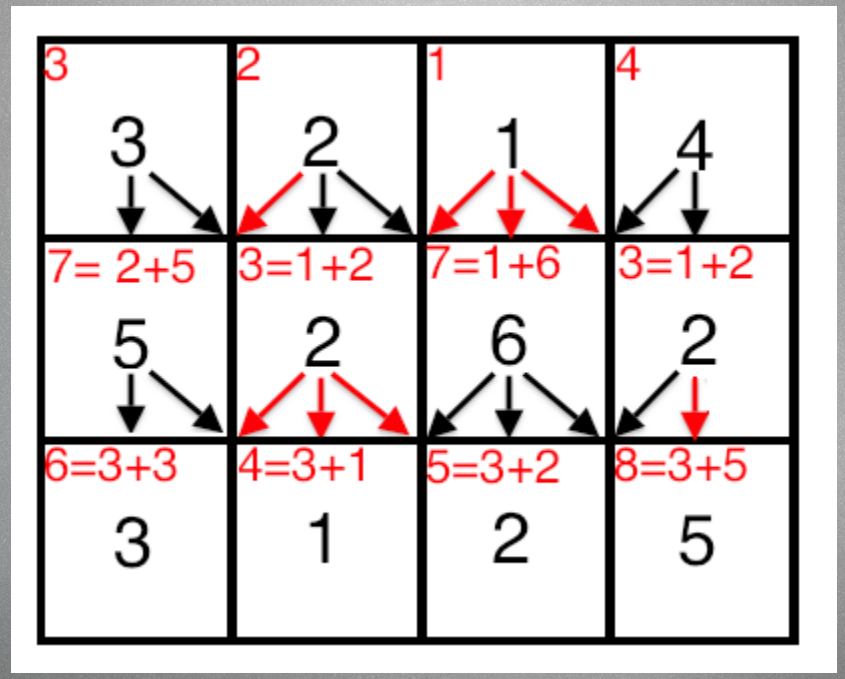
- GPS, vehicle routing and navigation
- Detecting arbitrage situations in FX
- RIP „Routing Information Protocol”
- This is a distributed algorithm
  - 1.) Each node calculates the distances between itself and all other nodes and stores this information as a table
  - 2.) Each node sends its table to all adjacent nodes
  - 3.) When a node receives distance tables from its neighbors, it calculates the shortest routes to all other nodes and updates its own table to reflect any changes



Avidan-Shamir method

- When we want to shrink an image for example in the browser or on a smartphone without distortion
- We want to make sure the image will not deform
- We have to eliminate the least significant bit strings
- We set up an „energy function”: and remove the connected string of pixels containing the least energy
- Photoshop, GIMP use it
- We build a huge graph: vertices are the pixels and the edges are pointing from every vertex to its downward 3 neighbours
- The energy function determines what the edge weights will be
- It’s acyclic: we can use topological order shortest path to find the string of pixels to be removed

Avidan-Shamir method



Longest path problem

- Problem of finding a simple path of maximum length in a given graph
- No polynomial time algorithm !!!
- It is an NP-hard problem
- It has a linear time solution for directed acyclic graphs (DAG) which has important applications in finding the critical path in scheduling problems

Longest path problem

- Problem of finding a simple path of maximum length in a given graph
- No polynomial time algorithm !!! NP-hard problem
- It has a linear time solution for directed acyclic graphs (DAG) which has important applications in finding the critical path in scheduling problems
- We just have to negate the edge weights and run shortest path algorithm
- We have to use Bellman-Ford algorithm because negative edges can occur
- Application: Parallel job scheduling problem
- Given a set of jobs with durations and precedence constraints, schedule the jobs - by finding a start time to each - so as to achive the minimum completion time, while respecting the constraints

CPM: critical path method

- The method was first used between 1940 and 1943 in the Manhattan project
- Problem formulation: we want an algorithm for scheduling a set of project activities so that the total running time will be as minimal as possible
- The algorithm needs
- A list of all activities required to complete the project
- The time (duration) that each activity will take to complete
- The dependencies between the activities

CPM: critical path method

- We create an edge weighted DAG
- Add edges with 0 weight for each precedence constraint
- We have to find the longest path in order to solve the problem
- There are no cycles in such a graph