Some exercises on MSTs:

1) Show how to find a maximum spanning tree of a graph, that is, a spanning tree of largest total weight

2) We know that if the weights of the edges are all distinct then $\exists$ a unique MST; show that the second-best MST, that is, the spanning tree of second-smallest total weight, is not necessarily unique

---

Shortest Paths

Definitions & terminology

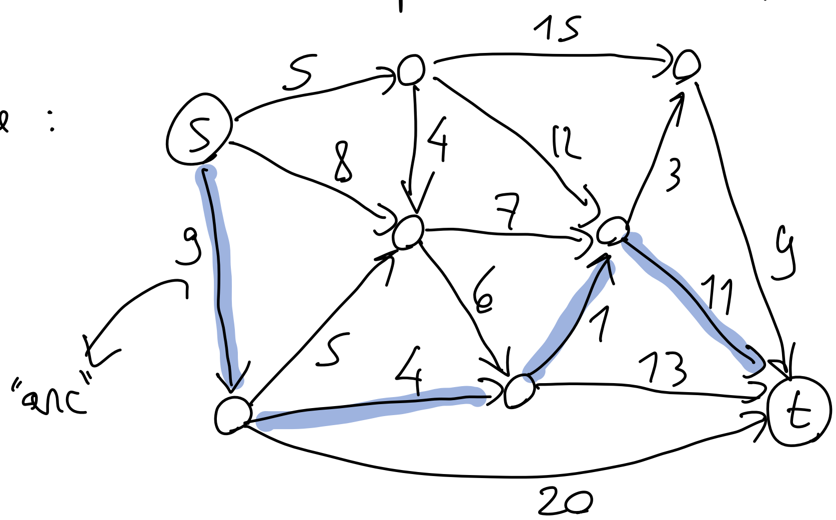
Given a weighted graph, the length of a path $P = v_1, v_2, \dots, v_k$ is defined as $len(P) = \sum_{i=1}^{k-1} w(v_i, v_{i+1})$

A shortest path from a vertex u to vertex v is a path with minimum length among all u-v paths

The distance between two vertices s and t, denoted $dist(s,t)$ is the length of a shortest path from s to t; if there is no path at all from s to t then $dist(s,t) = +\infty$

Problem: given a directed, weighted graph and a source vertex $s \in V$ and a destination $t \in V$, compute the shortest path from s to v

Example:



$$\mathrm{dist}(s,t) = 25$$

obs.: in directed graphs, $\underbrace{\mathrm{dist}(v,u)}_{\text{in general}} \neq \mathrm{dist}(u,v)$

Applications: – road networks (Google Maps)
– routing in networks (e.g. Internet)
– robots navigation

We'll solve this problem:

Single-Source Shortest Paths (SSSP)

input: a directed, weighted graph G with edge weights $w: E \to \mathbb{R}$ and a source vertex $s \in V$

output: $\text{dist}(s, v)$ $\forall$ vertex $v \in V$

Comments:
- no algorithms are known that run asymptotically faster than the best SSSP algorithm in the worst case
- we'll work with directed graphs, but all the algorithms that we'll see can be adapted easily for undirected graphs

We'll first solve a special case: nonnegative edge weights $w: E \to \mathbb{R}_{\geq 0}$

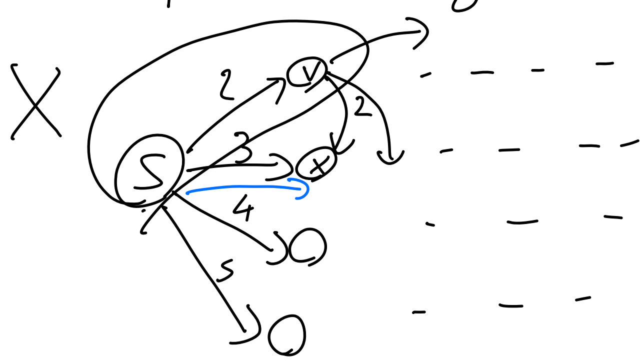
We've already solved a special case of this when all weights are $w = 1$ $\to$ solved in linear time using BFS

$$\circ \xrightarrow{3} \circ \qquad \xrightarrow{\text{replace}} \qquad \circ \xrightarrow{1} \circ \xrightarrow{1} \circ \xrightarrow{1} \circ$$

and then run BFS

Does it work? What if $w = 3,14|15\ldots$?
there is a bigger issue: the size of the graph can be much bigger than the size of the starting graph $\Rightarrow$ BFS takes linear time in the "bigger" graph, and this is not necessarily linear time in the size of the original graph!

Intuition for a new algorithm:



the arc (S,V) must be the shortest path from S to V since the first segment of any other path is already longer; a similar reasoning works in the next steps

Dijkstra's algorithm (1956)

greedy algorithm, very similar to Prim

Dijkstra (G, s)

input: directed G, $s \in V$, $w: E \to \mathbb{R}_{\geq 0}$

output: $len(v) = dist(s, v) \quad \forall v \in V$

$X = \{s\}$

$len(s) = 0$

$len(v) = +\infty$ \\ initial estimated distance

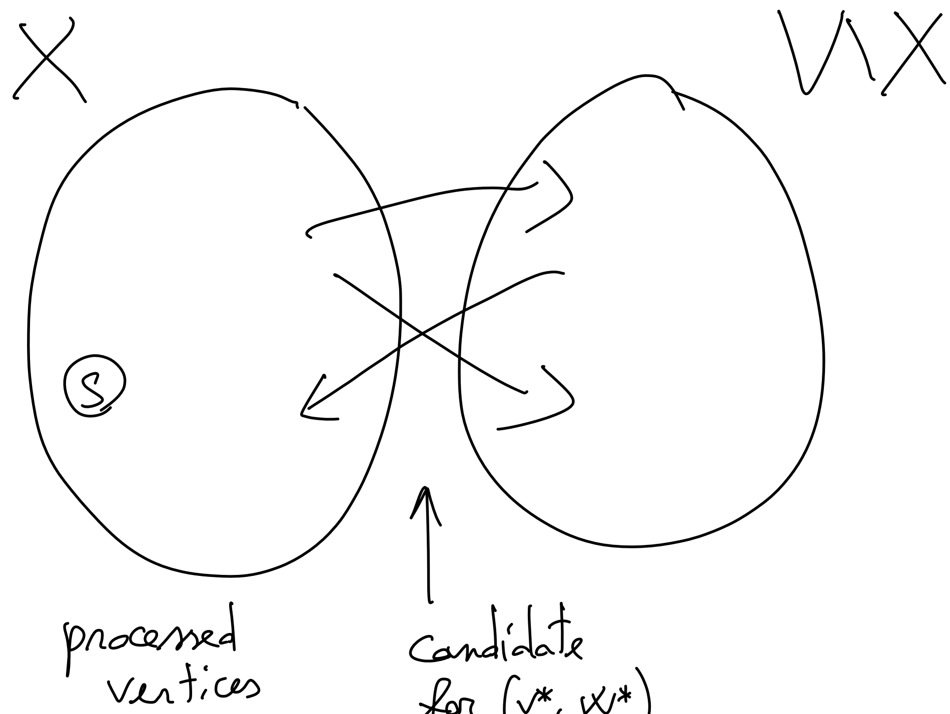
while there is an edge (v, w) with $v \in X$ and $w \notin X$ do

  (v^*, w^*) = such an edge minimizing $len(v) + w(v,w)$

  add w^* to X

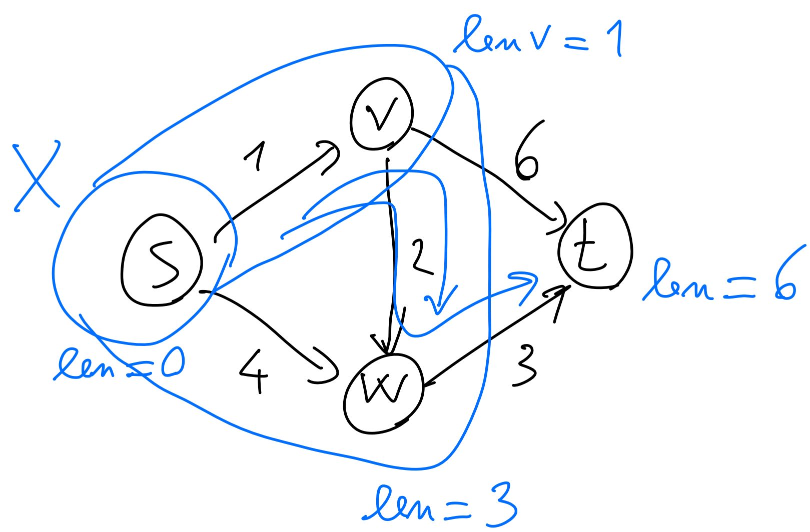
  $len(w^*) = len(v^*) + w(v^*, w^*)$

Exercise: modify Dijkstra's algorithm so as to compute the shortest paths themselves (and not just their lengths)



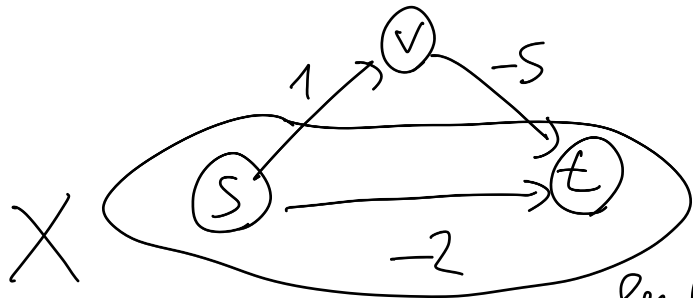
at each iteration a new node gets processed: w*

Example:



Praise of Dijkstra's alg.: in each iteration it irrevocably and myopically estimates the shortest path distance to one additional vertex despite having so far looked at only a fraction of the graph!

Dijkstra's algorithm does <u>not</u> work on graphs with negative weights:



but len(t) = -4!

len(t) = -2

Correctness of Dijkstra's algorithm:

invariant: $\forall x \in X$, len(x) is dist(s,x)

by induction on $|X|$

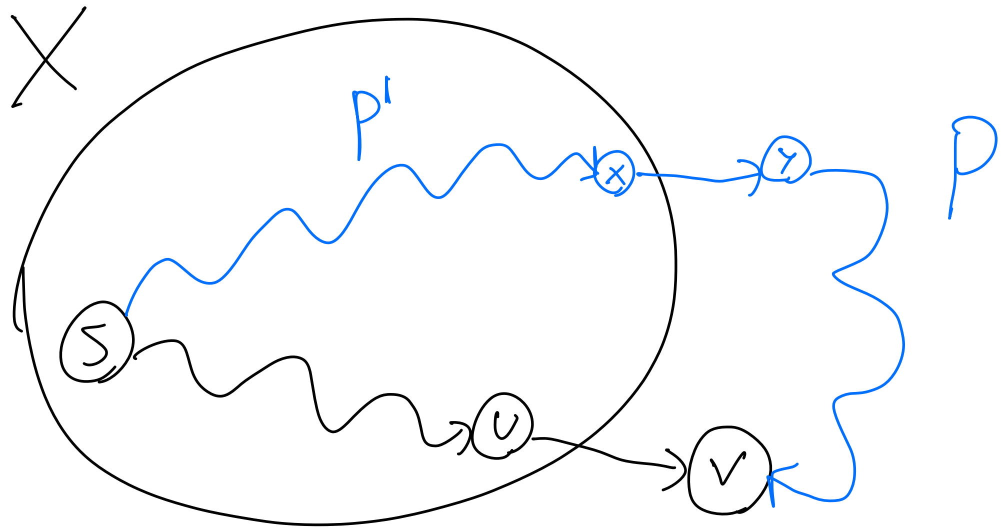
base case: $|X| = 1$ trivial

inductive hypothesis: invariant is true $\forall\ |X| = k \geq 1$

- let v the next vertex added to X, and (u,v) the arc
  - $\to w^*$
  - $\to (v^*, w^*)$

- the shortest path from s to u + (u,v) is a path from s to v of length $\pi(v) = \min\limits_{\substack{(u,v):\, u \in X \\ v \notin X}} \text{len}(u) + w(u,v)$

- consider any path P from s to v
  we'll show that P is not shorter than $\pi(v)$



let (x,y) be the first arc in P that traverses X
and let P' be the sub-path from s to x

$$\mathrm{len}(P) \underset{\text{w is nonnegative}}{\geq} \mathrm{len}(P') + w(x,y) \underset{\text{ind. hyp.}}{\geq} \mathrm{len}(x) + w(x,y) \geq$$

$$\underset{\text{def. of } \pi(v)}{\geq} \pi(y) \underset{\text{Dijkstra selected } v \text{ instead of } y}{\geq} \pi(v)$$

Complexity: $O(mn)$

Exercise: write an implementation of Dijkstra's alg. with heaps