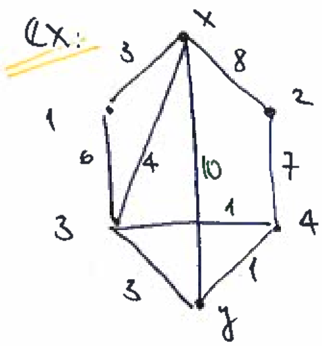
Disc Math - §22 - Shortest Path and Minimal Spanning tree.

Shortest path problem: Given a simple, connected, weighted graph (positive weights), a path exists between any two nodes x and y. Find the path $x \to \dots \to y$ with minimum weight. → Dijkstra's algorithm

ex:



Adjacency matrix: ↙ (Modified adjacency matri

$$
\begin{array}{c} \\ x \\ 1 \\ 2 \\ 3 \\ 4 \\ y \end{array}
\begin{array}{c}
\begin{array}{cccccc} x & 1 & 2 & 3 & 4 & y \end{array} \\
\begin{pmatrix}
\infty & 3 & 8 & 4 & \infty & 10 \\
3 & \infty & \infty & 6 & \infty & \infty \\
8 & \infty & \infty & \infty & 7 & \infty \\
4 & 6 & \infty & \infty & 1 & 3 \\
\infty & \infty & 7 & 1 & \infty & 1 \\
10 & \infty & \infty & 3 & 1 & \infty
\end{pmatrix}
\end{array}
$$

We will iteratively construct a set D.

(1) $D = \{x\}$

	x	1	2	3	4	y
d	0	3	8	4	∞	10
p	–	x	x	x	x	x

← current distance to x.
← predecessor in the path

(2) Add the node closest to x: $D = \{x, 1\}$

$D = \{x, 1\}$

	x	1	2	3	4	y
d	0	3	8	4	∞	10
p	–	x	x	x	x	x

← current distance to x, paths could go throu
← predecessor in the shortest path.

(3) Add 3: $D = \{x, 1, 3\}$

	x	1	2	3	4	y
d	0	3	8	4	5	7
p	–	x	x	x	3	3

(4) Add 4: $D = \{x, 1, 3, 4\}$

	x	1	2	3	4	y
d	0	3	8	4	5	6
p	–	x	x	x	3	4

(5) Add y:

$D = \{x, 1, 3, 4, y\}$

	x	1	2	3	4	y
d	0	3	8	4	5	6
s	–	x	x	x	3	4

Stop: Shortest distance =

Shortest path: y, p(y) = 4, p(4) = 3, p(3) = x

$x \to 3 \to 4 \to y$

Disc Math - §22 - Shortest path and Minimal Spanning Tree

Dijkstra's shortest path algorithm in general.

1. Build a set D. Initially $D = \{x\}$.
2. keep track of the shortest distance $x \to \dots \to z$ for $z \notin D$ with paths only going through nodes currently in D. keep track of the predecessor of z on the shortest path.
3. Grow D by adding the node outside it with current shortest distance to x. Recompute all distances when the new node is added.
4. Stop when y is added to D. Adding further nodes clearly will not produce a path $x \to \dots \to y$ any shorter.

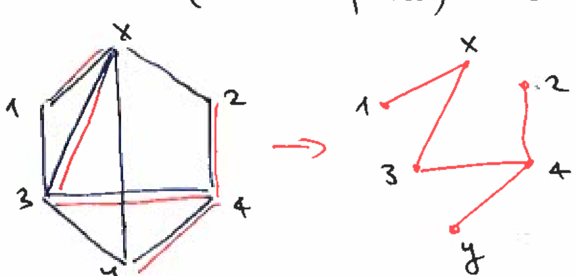
rem: i) Dijkstra's algorithm does not look at the whole graph at once to pick out the overall shortest path; it is a greedy algorithm - it does what seems best based on its limited immediate knowlegde (D).

ii) It is $\Theta(n^2)$.

Minimal Spanning Tree Problem.

Def: A spanning tree for a connected graph is a unrooted tree whose set of nodes coincides with the set of nodes for the graph and whose arc are (some of the) arcs of the graph.

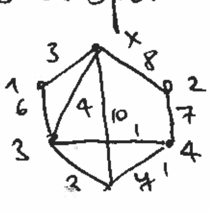
Ex:



A spanning tree connects all the nodes on the graph with no extra arcs.

Def: For a simple, connected, weighted graph a minimal spanning tree is a spanning tree with a minimal weight.

Ex:



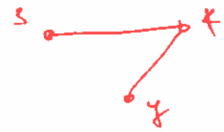
Start with an arbitrary node, say $P = \{3\}$.

Prim's Algorithm.

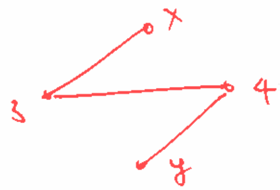
Disc Math - §22 - Shortest path and Minimal Spanning Tree

Add the node closest to D → $D = \{3, 4\}$

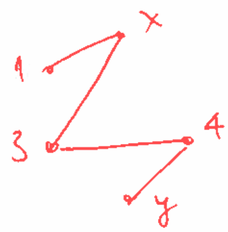
Add the node closest to D → $D = \{3, 4, y\}$



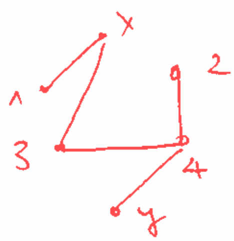
D → $D = \{3, 4, y, x\}$



D → $D = \{3, 4, y, x, 1\}$



D → $D = \{3, 4, y, x, 1, 2\}$



Stop.

rem: i) Because there might be ties between minimal distances, the MST may not be unique.

ii) Prim's algorithm is greedy and $\Theta(n^2)$. ☒

HW §7.3 p 591

3, 6, 8, 18, 20, 27.