Graphs : $g \equiv 2d+1$ (Moore graphs)

(connected) (girth) (diameter)

Claim :- Moore graphs are regular.

$\deg(u) = \deg(v)$ find a bijection

$\deg(u) = \deg(x) = \deg(v)$ between $N(u)$ and $N(v)$

(*) Moore graphs are regular.

Next : Moore graphs for $g=5$.

Degree sequence:

G : $v_1, v_2, \ldots, v_n$

$(\deg v_1, \deg v_2, \ldots, \deg v_n)$ and sort them in a non-increasing order.

Suppose you are given a non-increasing finite list of integers: Is it the degree sequence of some graph?

Claim:- $\sum_{v_i \in V} \deg(v_i) = 2 \times |E|$.

$\Rightarrow \sum_{\deg(v_i) \text{ is Even}} \deg(v_i) + \sum_{\deg(v_i) \text{ is Odd}} \deg(v_i) = \underline{2 \times |E|}$ (Even)

Suppose you are given a non-increasing finite list of integers: Is it the degree sequence of some graph?

Claim:- $\sum_{v_i \in V} \deg(v_i) = 2 \times |E|$.

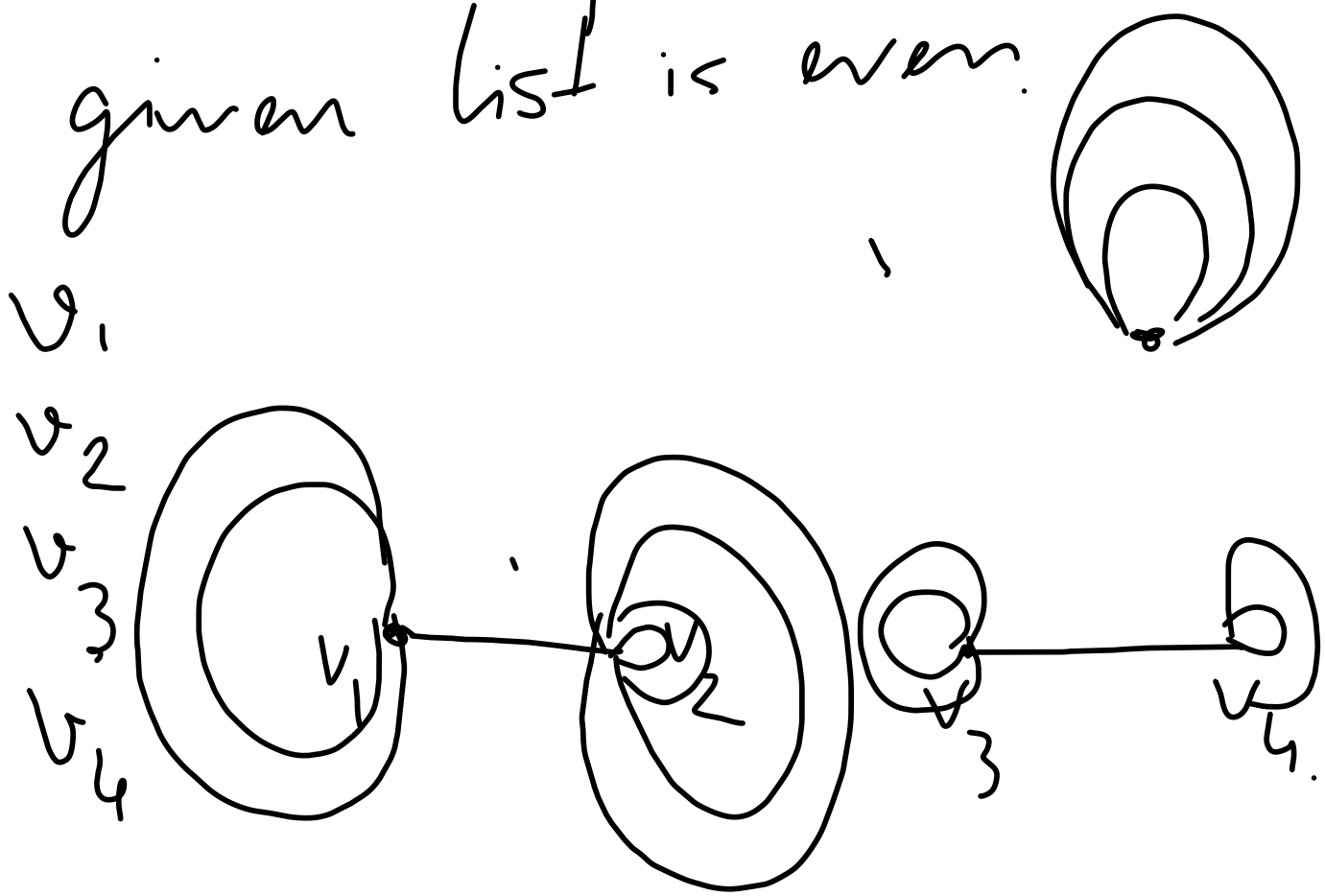
$\Rightarrow \sum_{\deg(v_i) \text{ is Even}} \deg(v_i) + \sum_{\deg(v_i) \text{ is Odd}} \deg(v_i) = \underline{2 \times |E|}$ (Even)

In any graph the no of vertices with odd degree is even.

Ans: the no of odd integers in the given list is even.

In any graph the no of vertices with odd degree is even.

Ans: the no of odd integers in the given list is even.

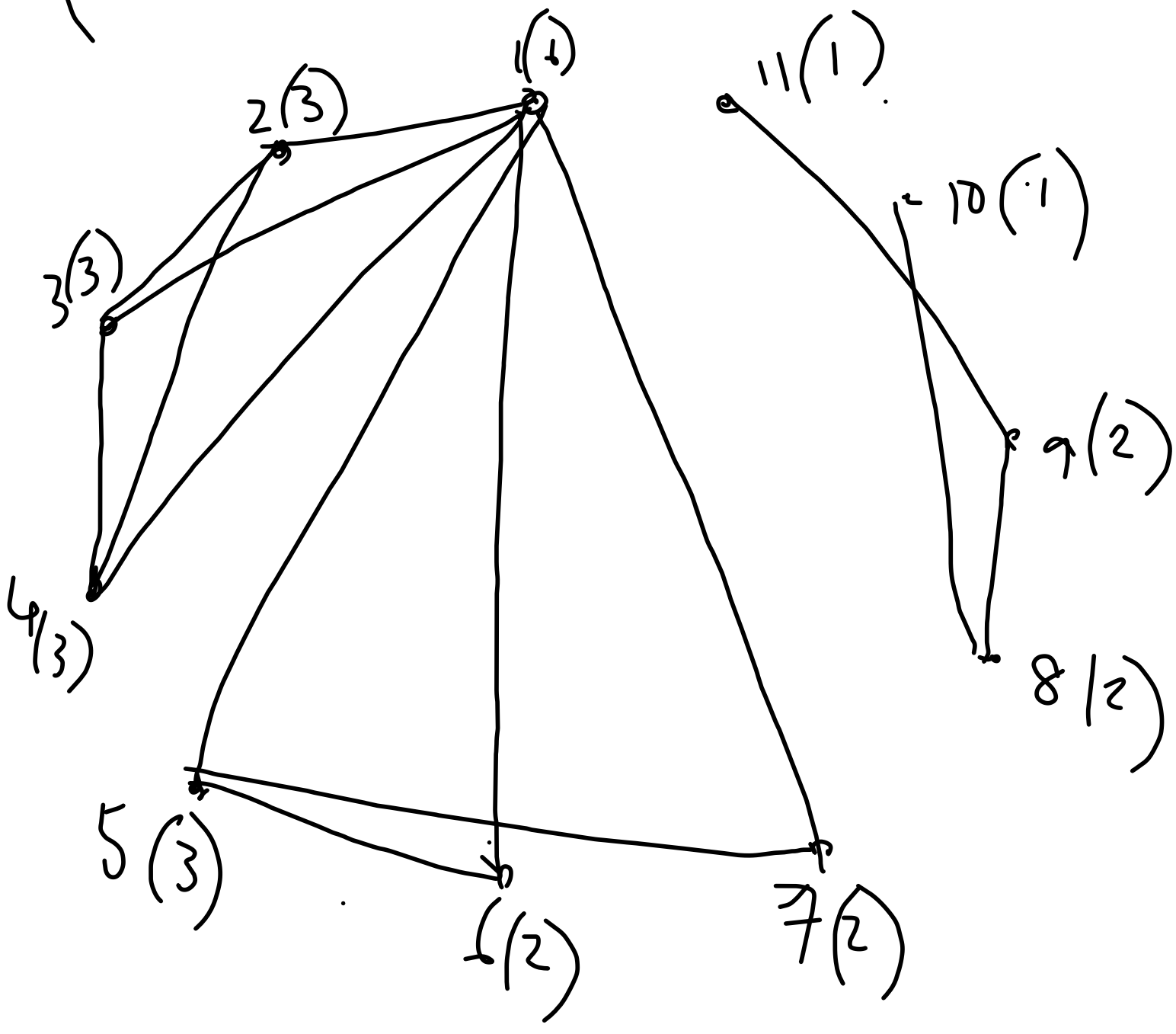


A degree sequence is called a graphic sequence if it is the degree sequence of a simple graph. (no loops, no parallel edges)

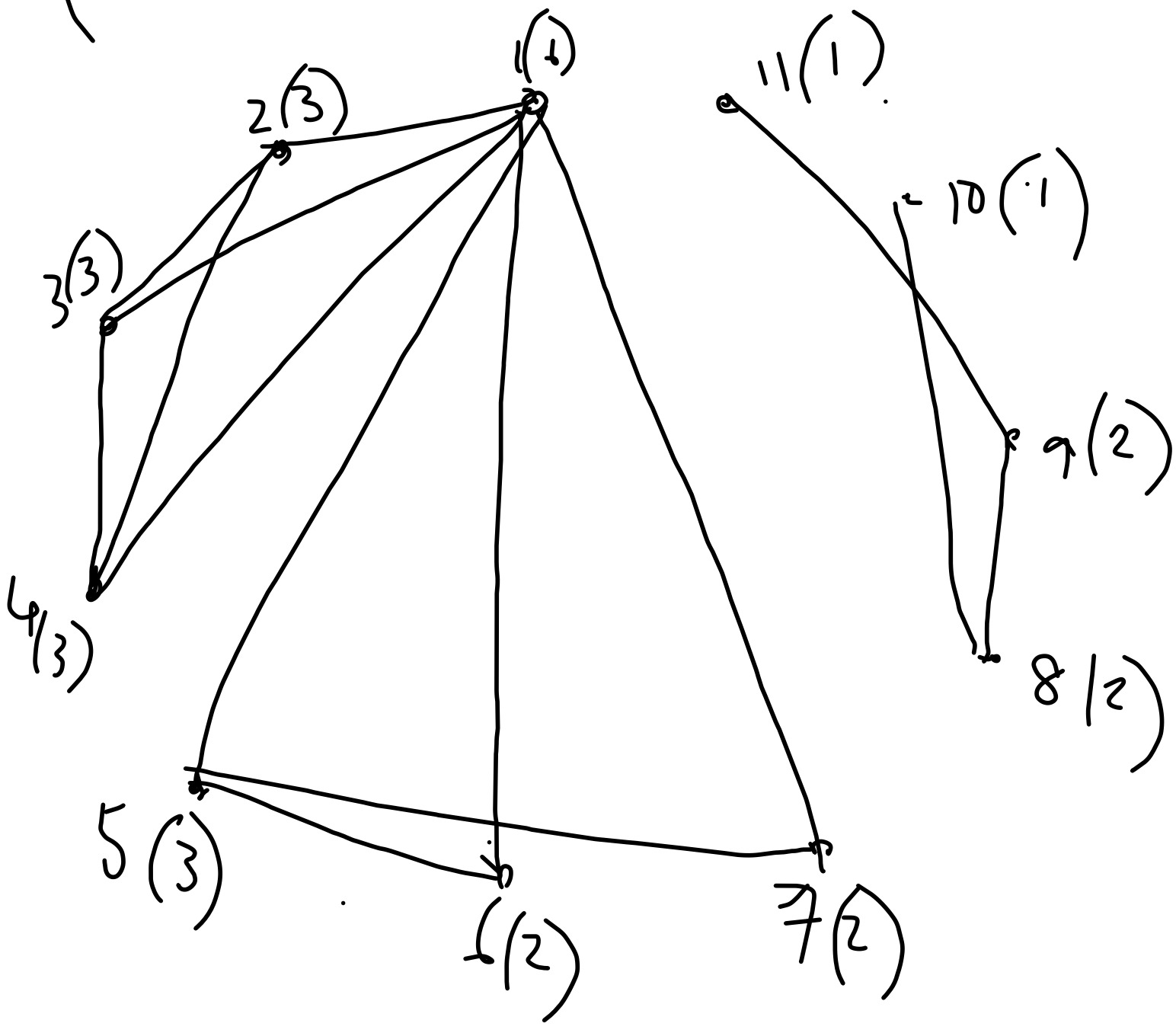
Q: When is a degree sequence a graphic sequence?

(6, 3, 3, 3, 3, 2, 2, 2, 2, 1, 1)

(6, 3, 3, 3, 3)



- $(6, 3, 3, 3, 3, 2, 2, 2, 2, 1, 1)$



$(6, 3, 3, 3, 3)$

$A = (S, t_1, t_2, \ldots, t_S, d_1, d_2 \ldots, d_n)$

$A' = (t_1 - 1, t_2 - 1, \ldots, t_{S-1}, d_1, d_2, \ldots d_n)$

$\hookrightarrow A'' \longrightarrow A''' \longrightarrow \ldots \emptyset$

A is a graphic sequence iff

A' is " "

Havel-Hakimi Algorithm. (Exc:- running time)

Q:- How to decide if a graph is connected or not?

Ans:- by definition:- if there are two vertices which are not joined by any path.

What is a smarter way! a polynomial time algorithm (if it exists).

Spanning Subgraph of a graph G is a subgraph which is obtained by edge deletion.

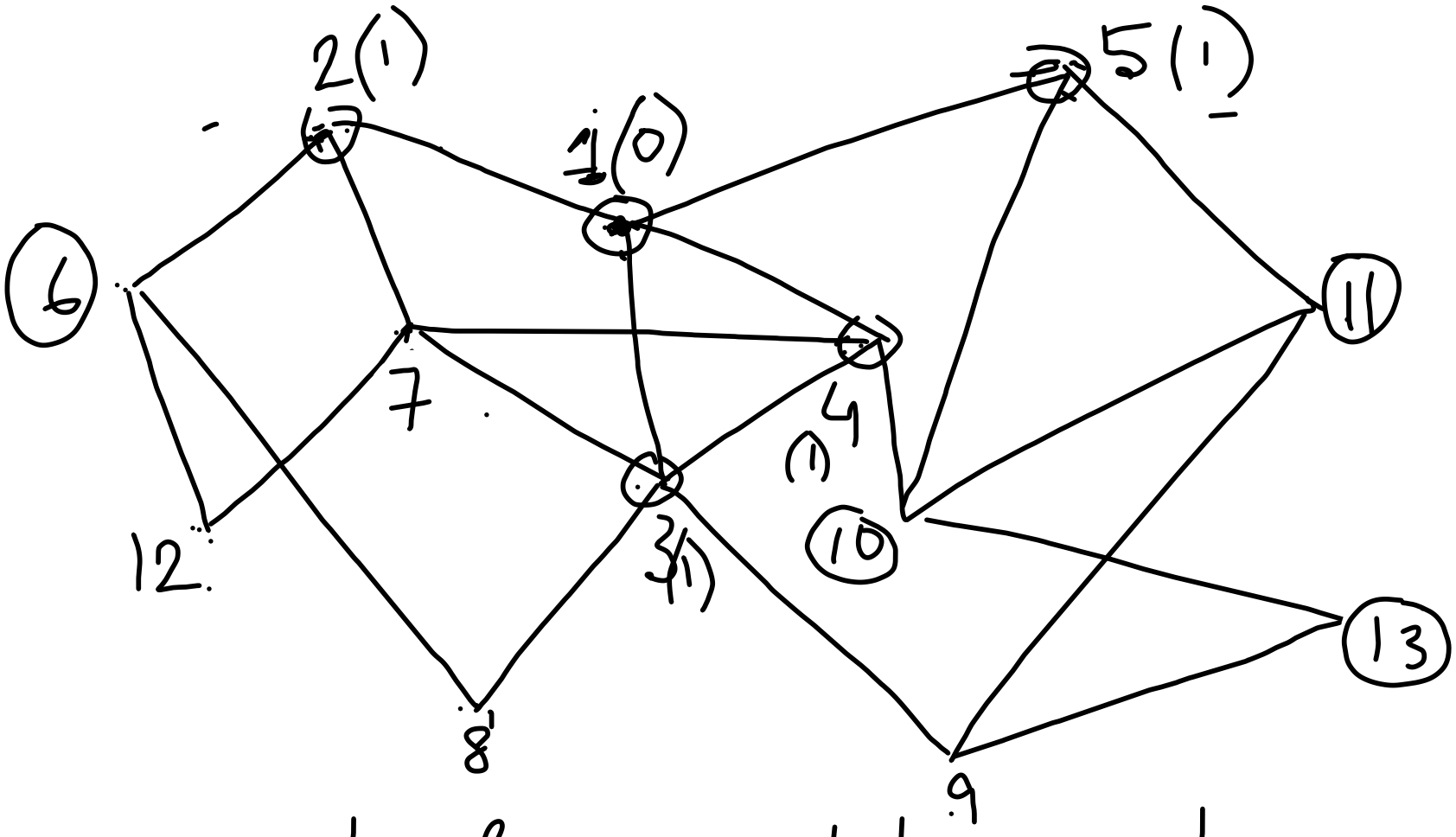
Spanning tree :— Spanning subgraph which is a tree.

(*) A graph is connected iff it has a spanning tree —

Q. How to find Spanning trees inside a graph.

(1) Breadth first search.

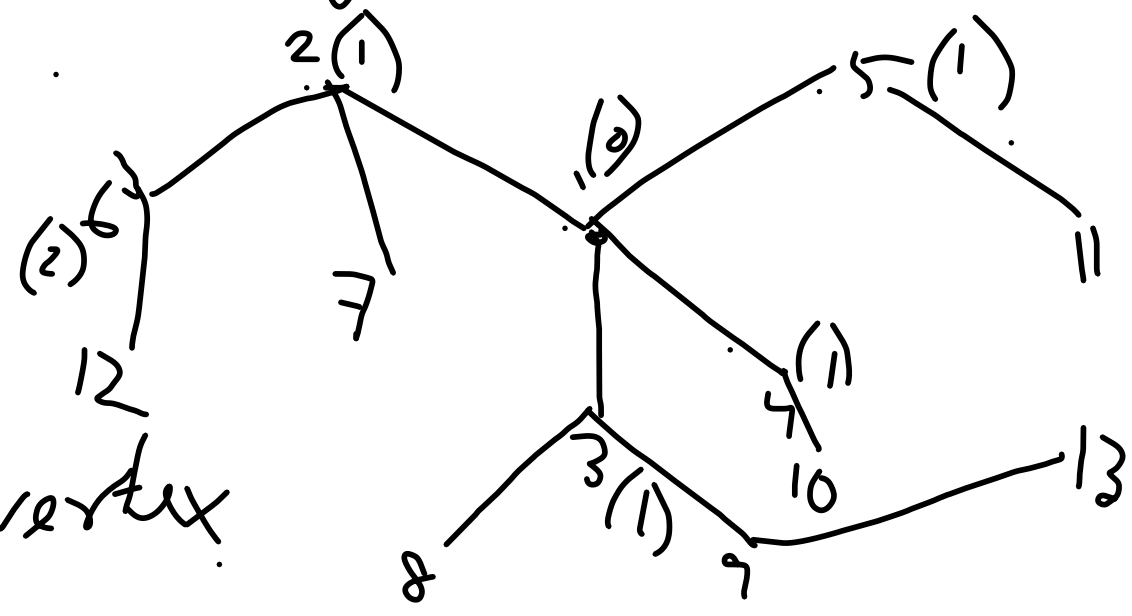
(2) Depth first search.



Breadth first search

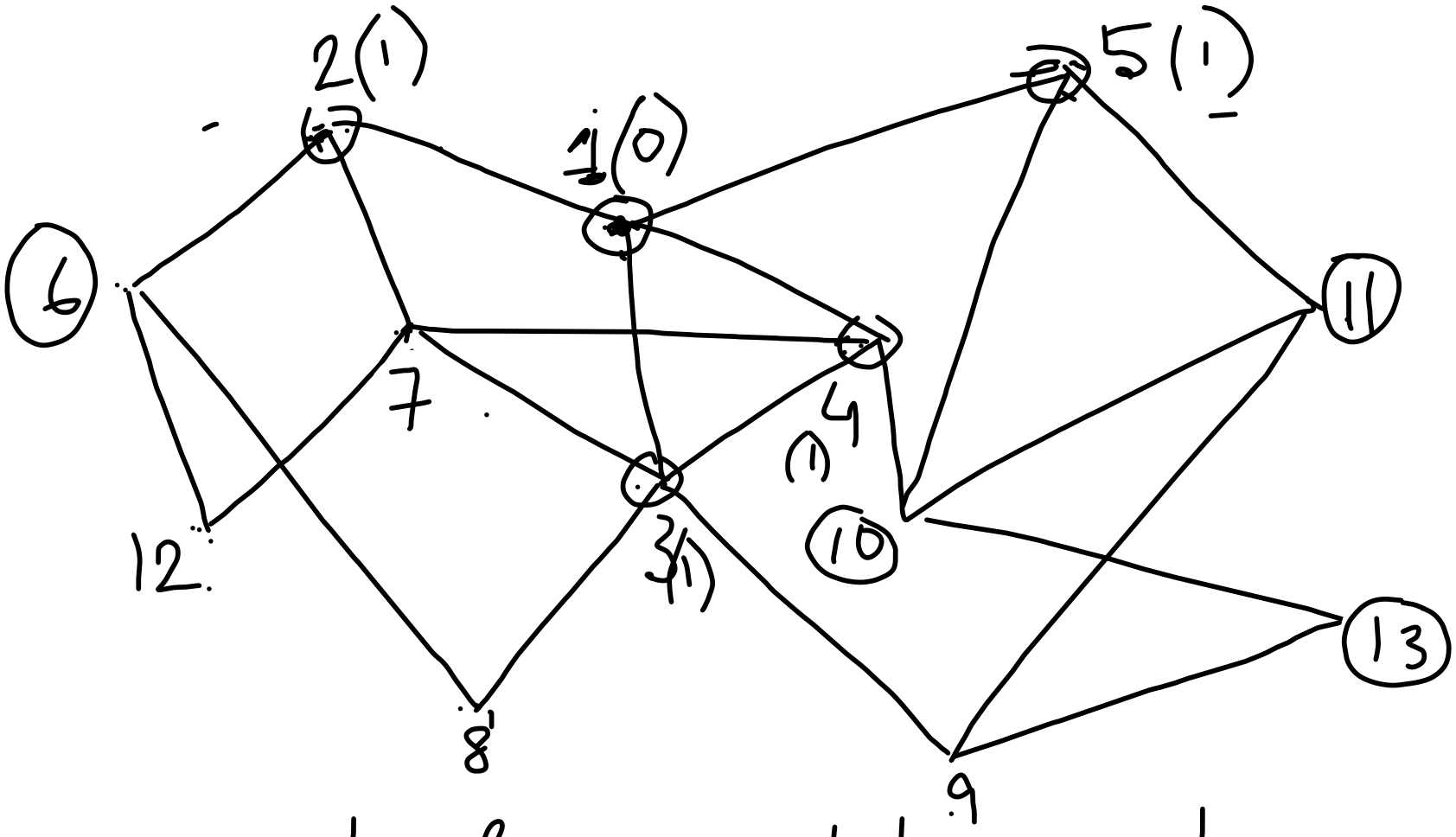
(search along the breadth and then proceed)

idea:- keep on adding edges until there is no cycle.



(spanning tree)

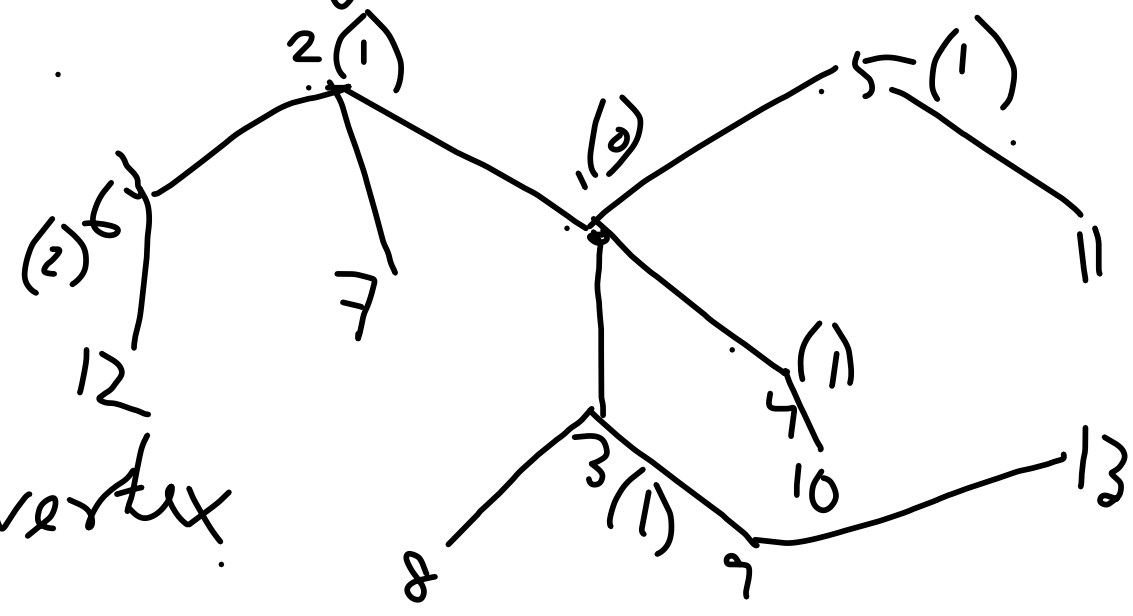
can also get the distance from the starting vertex.



Breadth first search

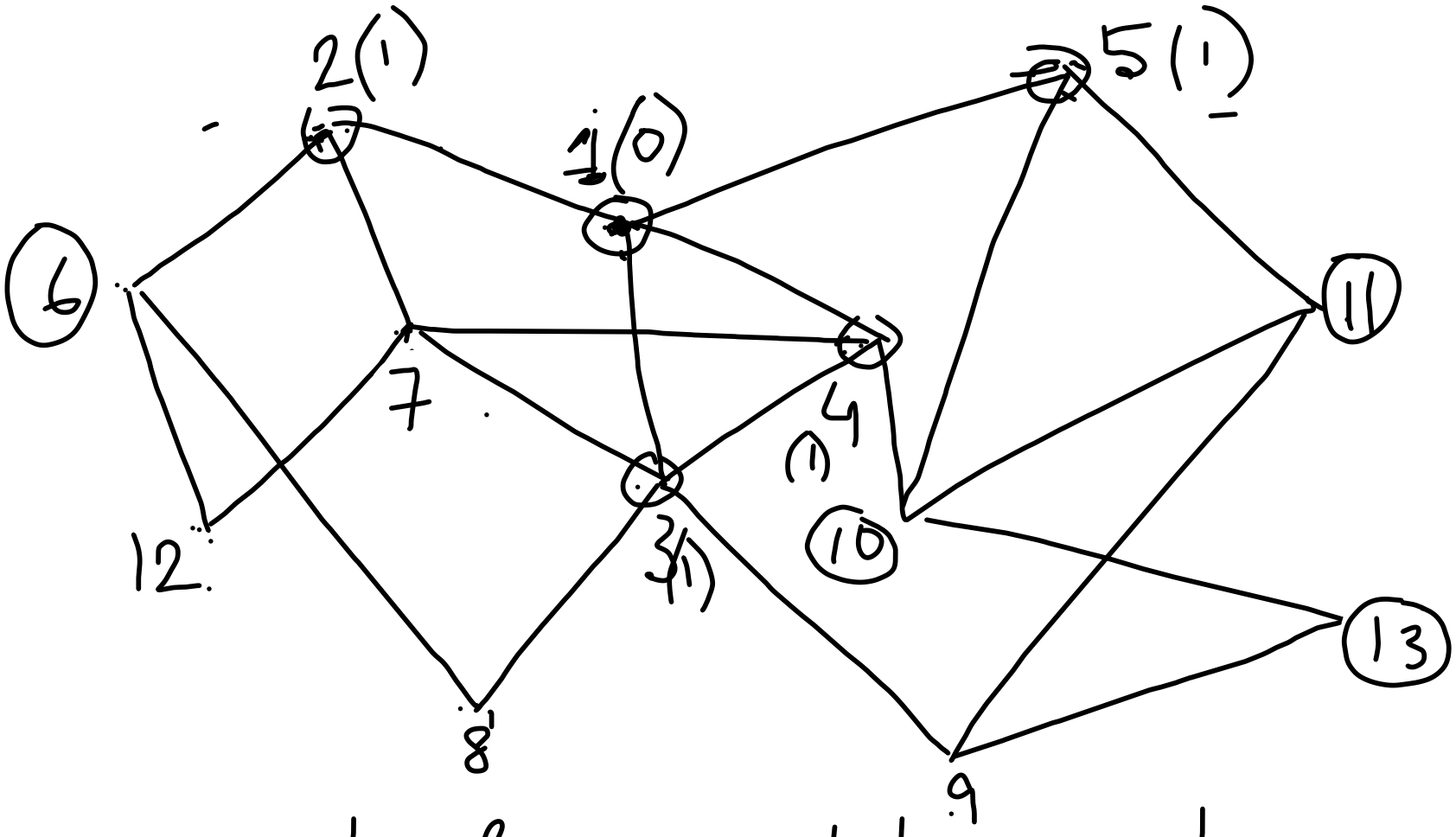
(search along the breadth and then proceed)

idea:- keep on adding edges until there is no cycle.



(Spanning tree)

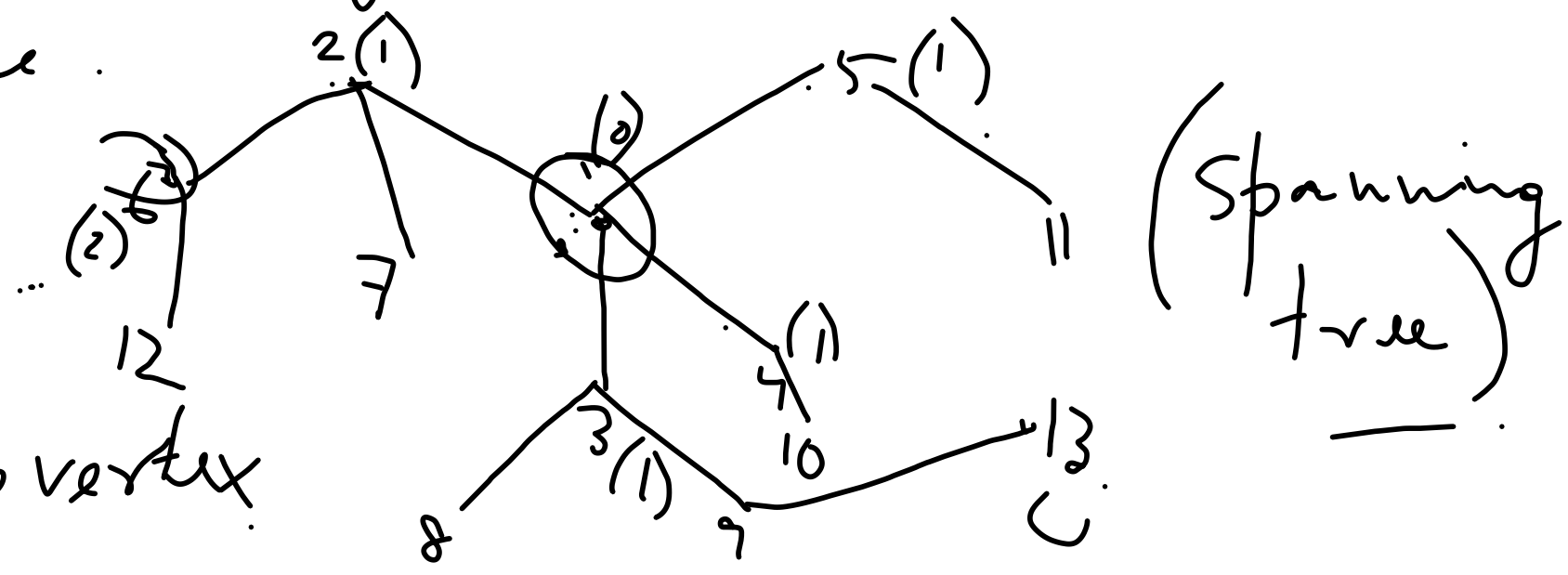
can also get the distance from the starting vertex.



Breadth first search.

(Search along the breadth and then proceed)

idea:- keep on adding edges until there is no cycle.



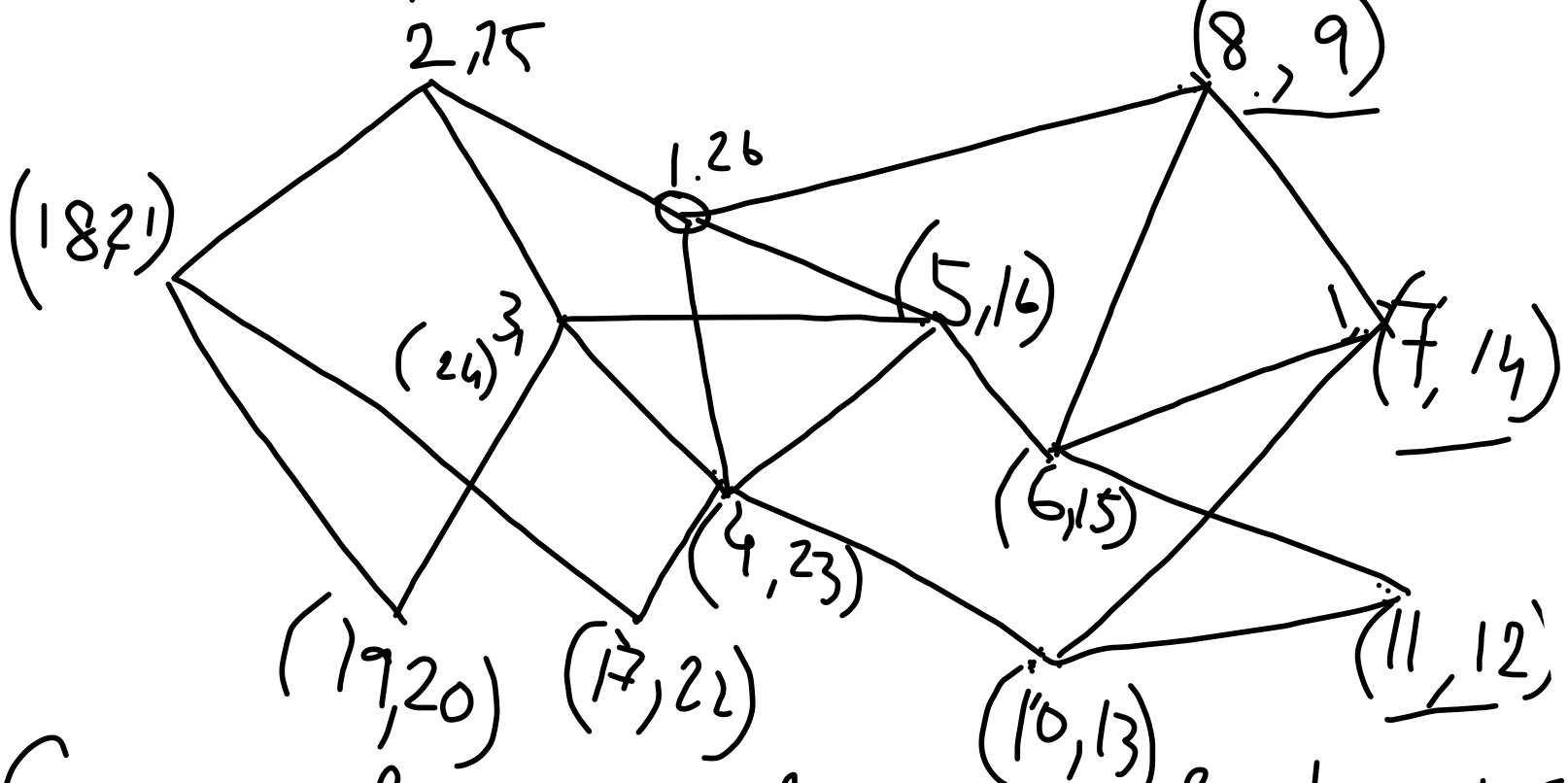
(Spanning tree)

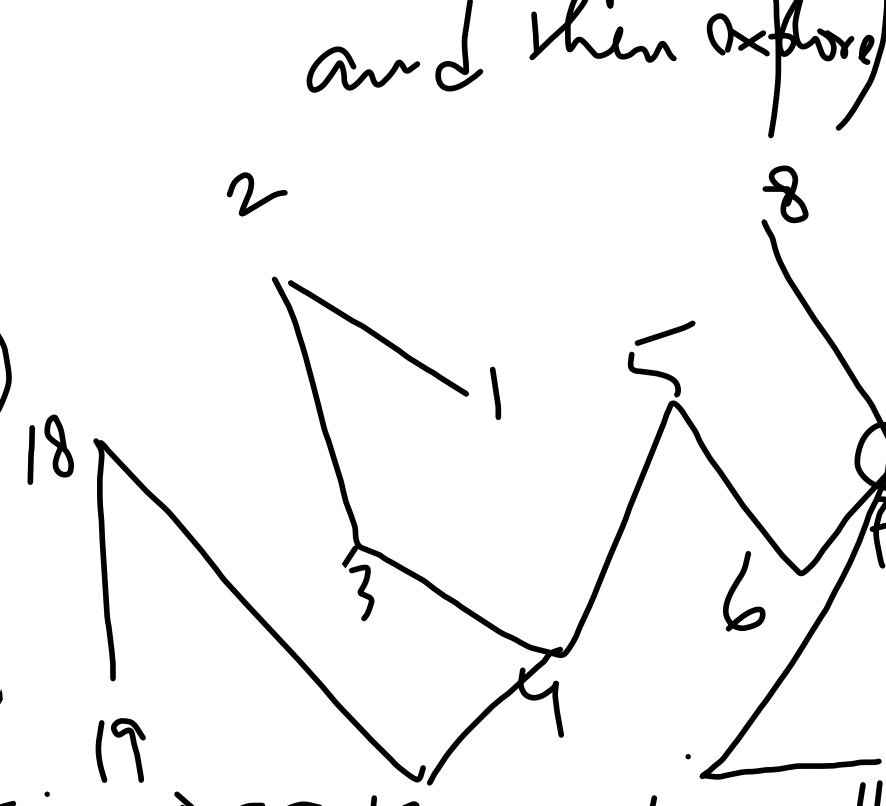
can also get the distance from the starting vertex.

Algorithm for detecting a cycle?

(union finding algorithm)

(2) Depth first search. idea → first go deep. and then explore



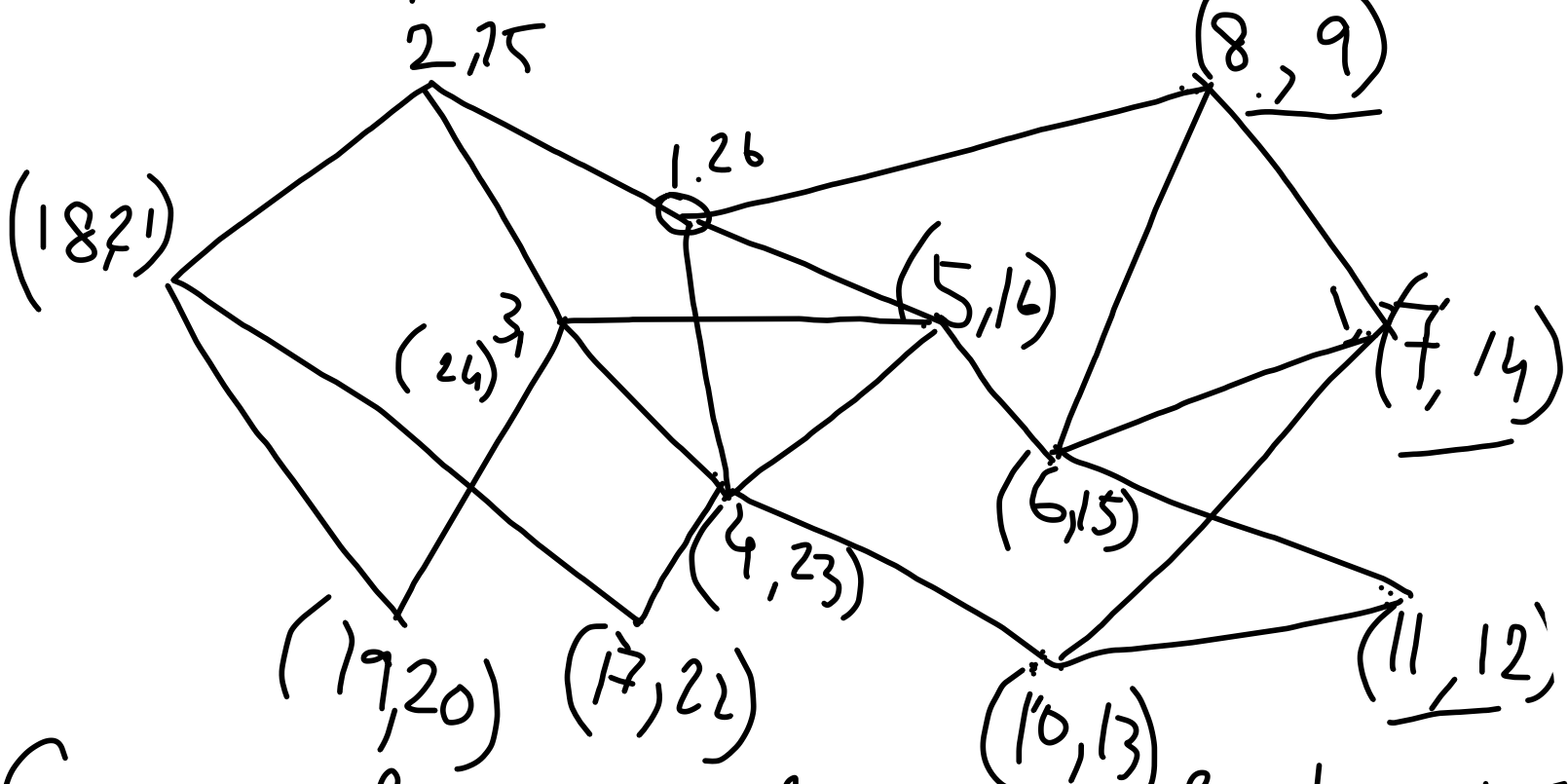


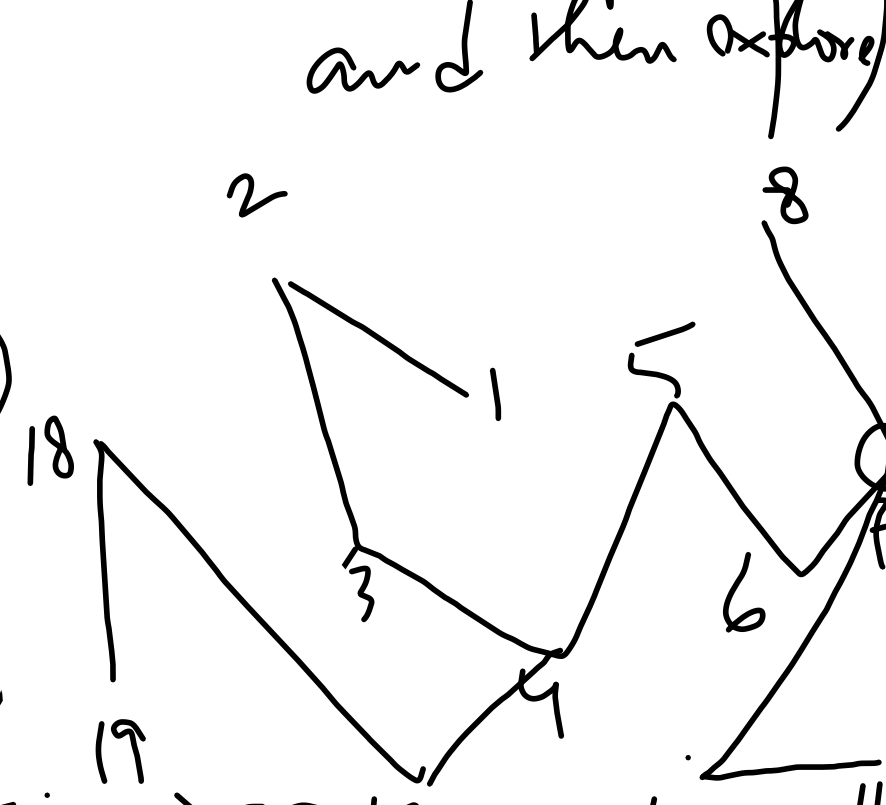
Can you find the diameter of a tree using DFS ? (Assignment)

Algorithm for detecting a cycle?

(union finding algorithm)

② Depth first search. idea → first go deep. and then explore

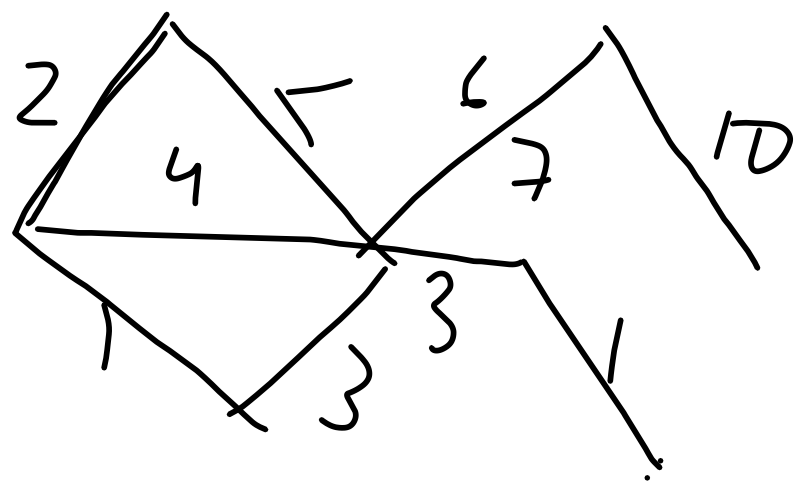




Can you find the diameter of a tree using DFS ? (Assignment)

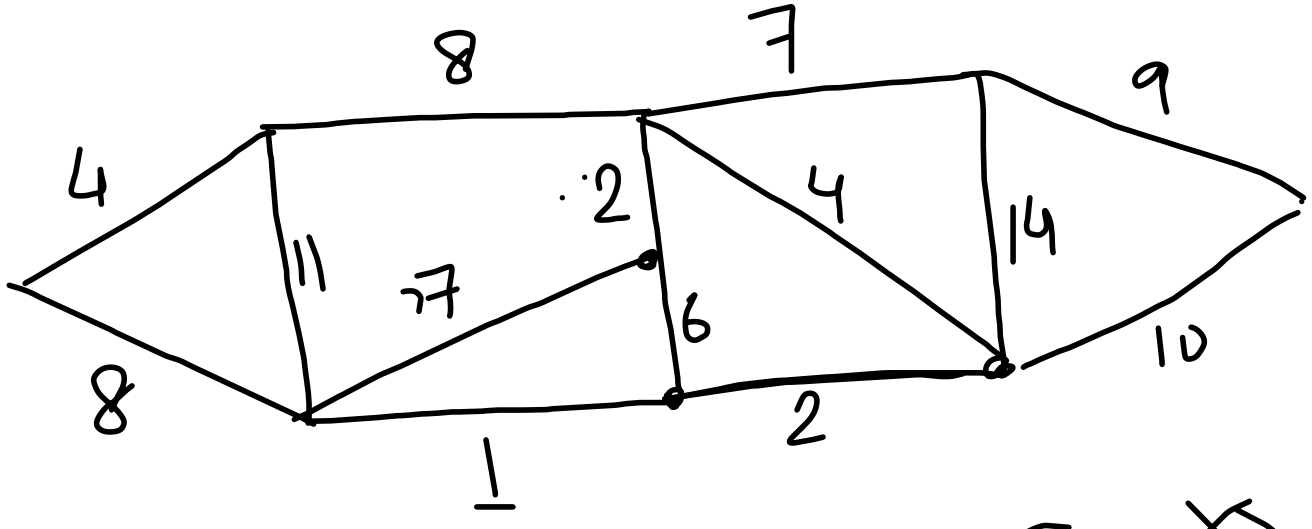
(*) If you run B.F.S on a tree twice, you can end up with the diameter. (Exc)

Suppose your graph has weights on the edges.

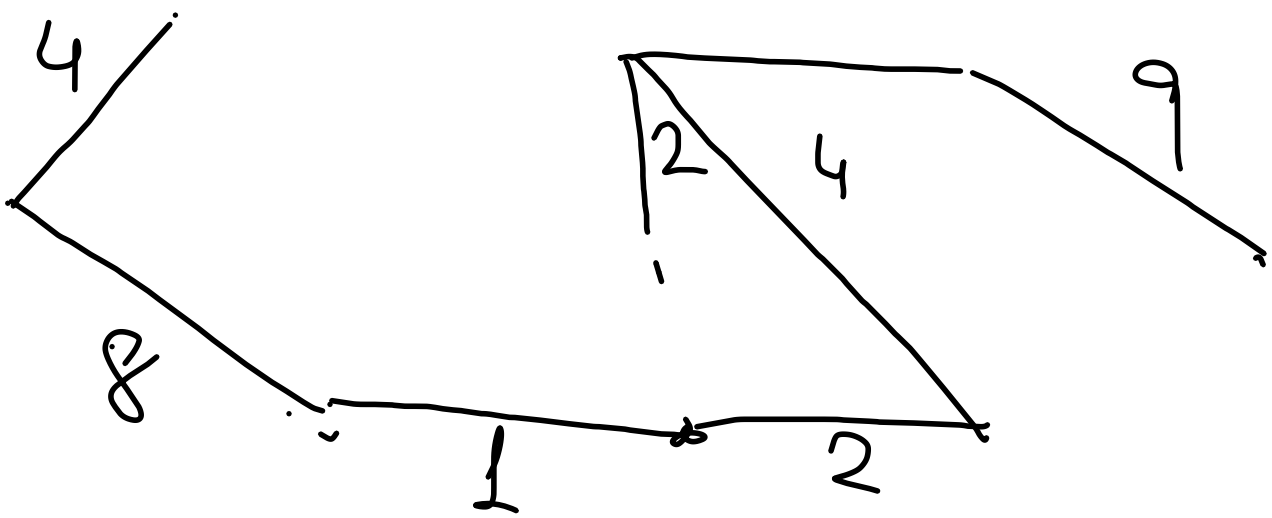


How can you find the minimum weight spanning tree?

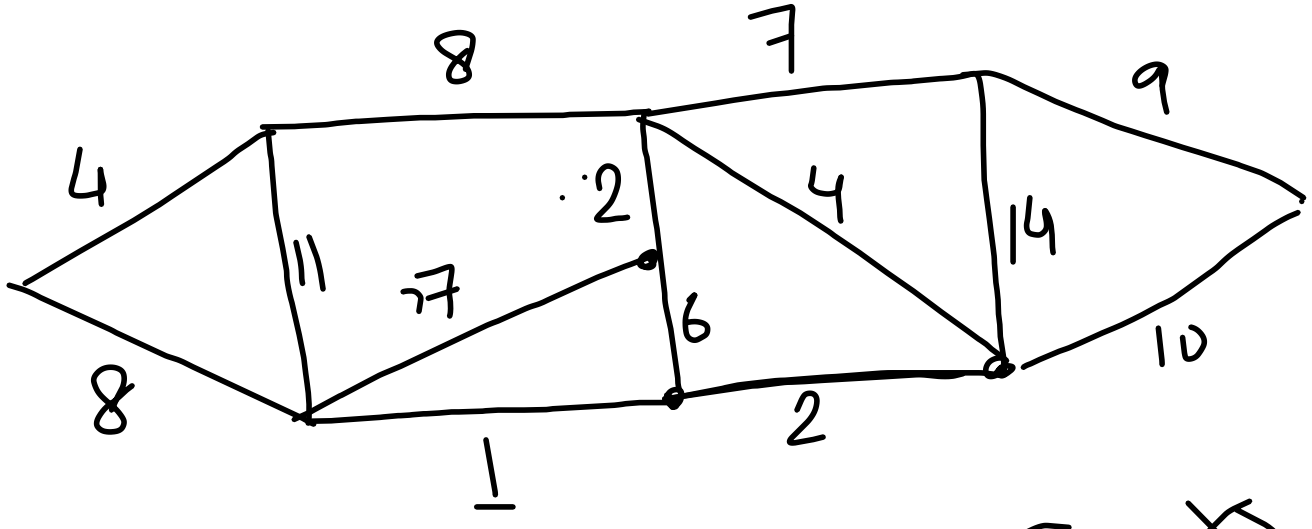
Kruskal's algorithm ! Prim's algorithm.



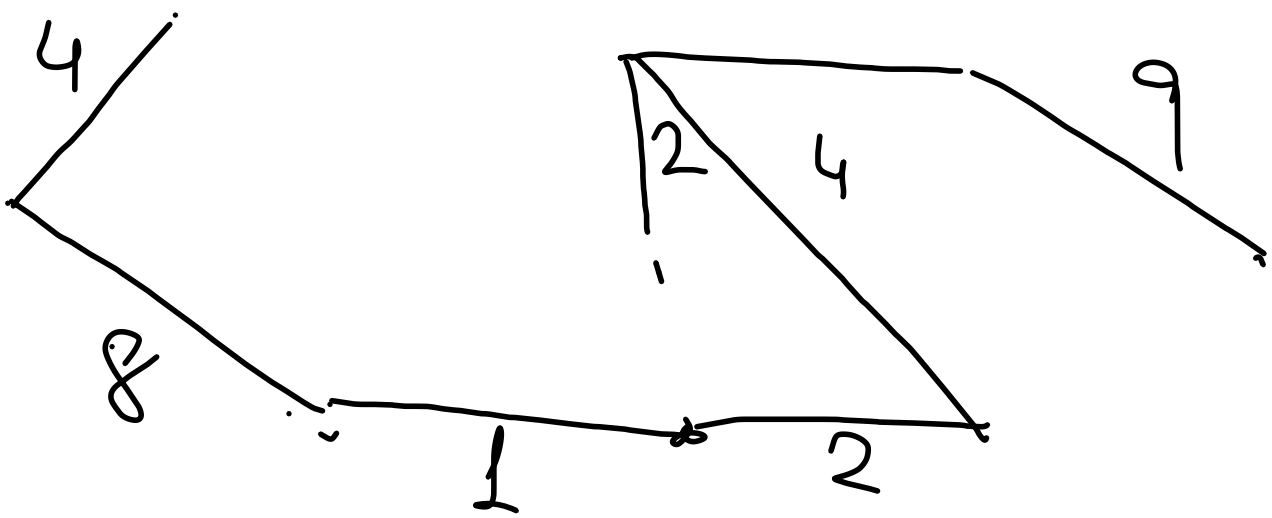
1 2 2 4 4 ~~6~~ ~~7~~ 7 8 ~~8~~ 9 ~~10~~ ~~11~~ ~~14~~



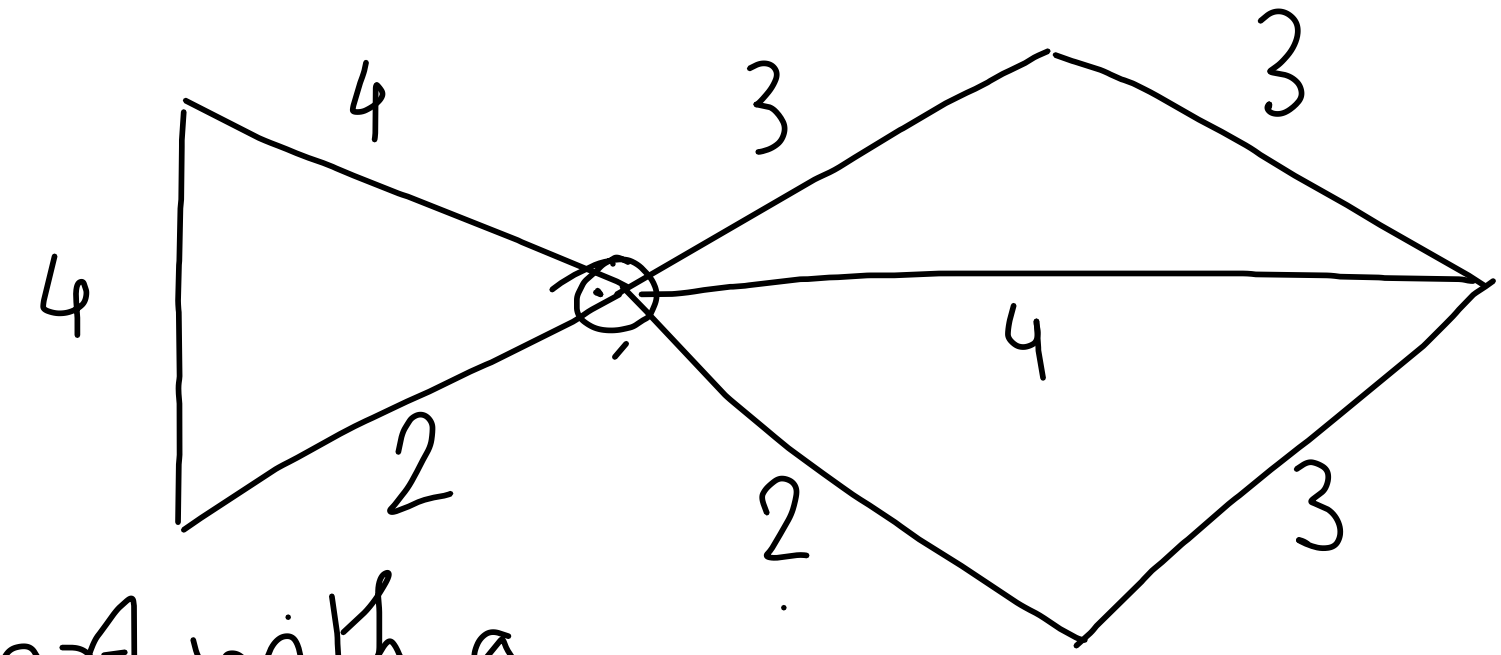
Kruskal's algorithm ! Prim's algorithm.



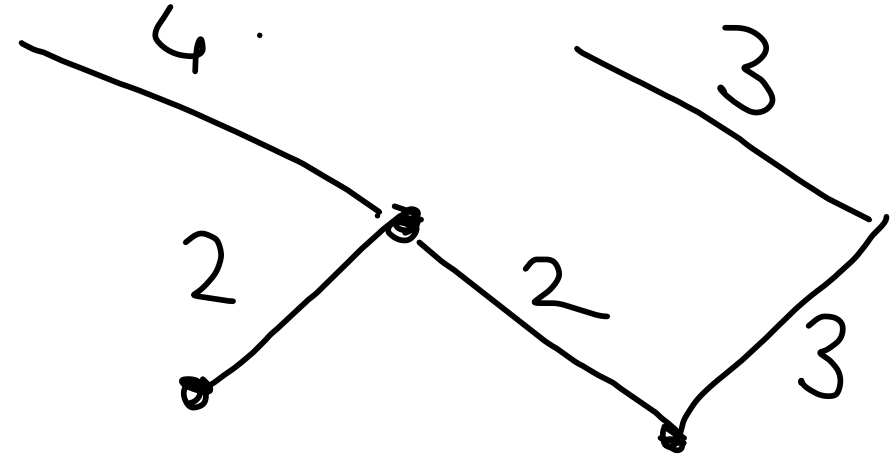
1 2 2 4 4 ~~6~~ ~~7~~ 7 8 ~~8~~ 9 ~~10~~ ~~11~~ ~~14~~



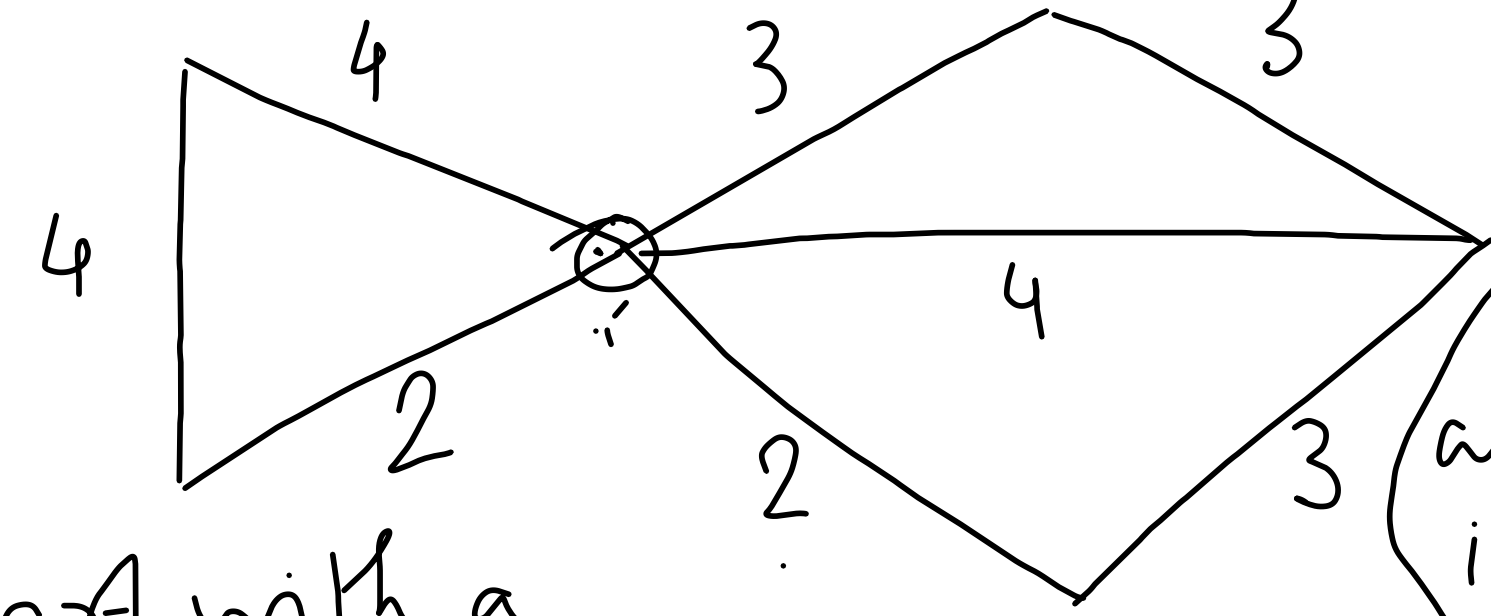
Prim's algorithm.



1. Start with a random vertex.
2. choose the adjacent edge with the lowest wt
3. Add it till it does not form a cycle.

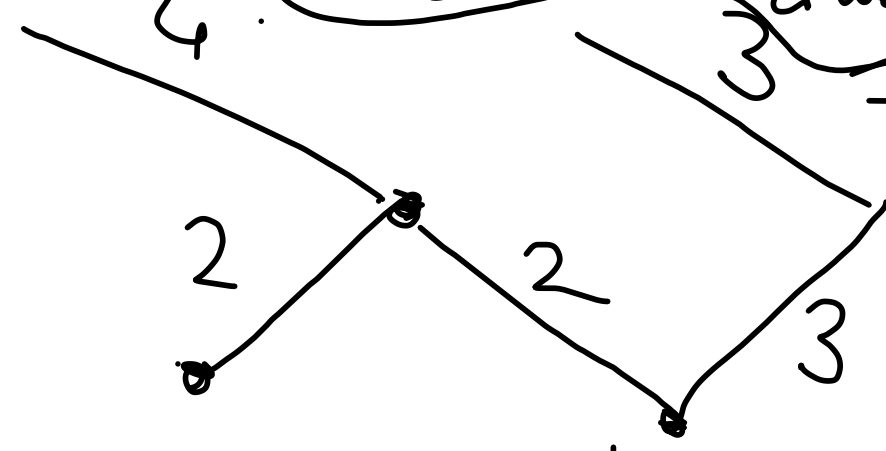


Prim's algorithm.



Greedy algorithms

will give an optimal solution if the underlying structure is a matroid



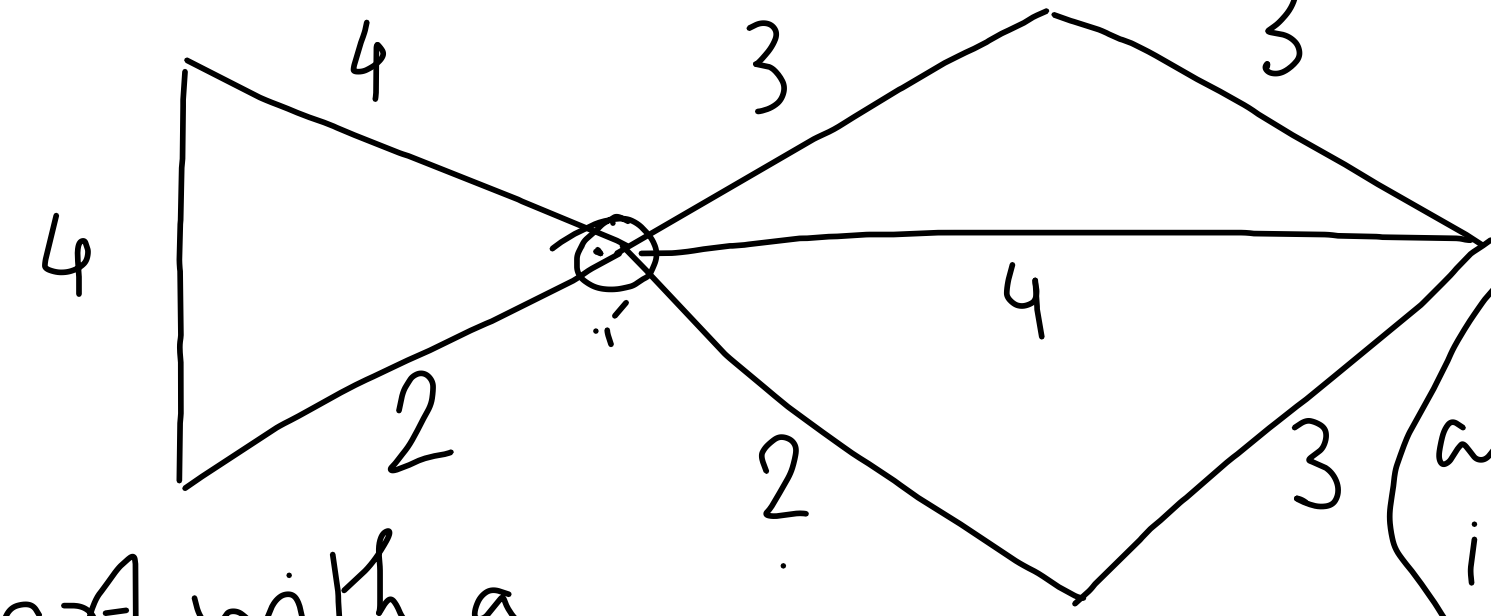
① Start with a random vertex.

② choose the adjacent edge with the lowest wt

③ Add it till it does not form a cycle.

ring times

Prim's algorithm.

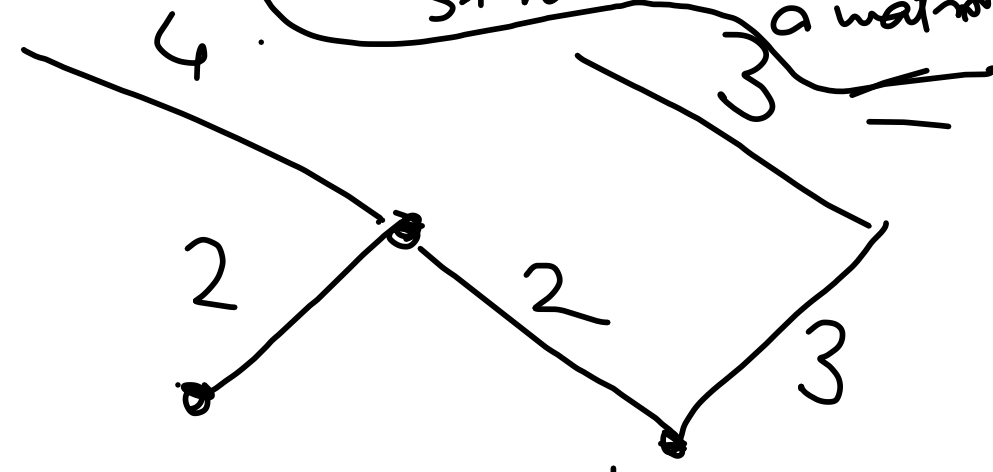


① Start with a random vertex.

② choose the adjacent edge with the lowest wt

③ Add it till it does not form a cycle.

Greedy algorithms will give an optimal solution if the underlying structure is a matroid



ring theory

(*) Union finding algorithm.
(to detect cycles)

Dijkstra's algorithm