Ex 12.1.2: a) Applying Dijkstra's algorithm in steps gives the following:

step 1:

vertex	distance
a	0
b	14
c	∞
f	∞
g	10
h	17
i	∞

step 2:

a	0
b	13
c	∞
f	∞
g	10
h	16
i	14

Step 3:

a	0
b	13
c	22
f	23
g	10
h	16
i	(14)

Step 4:

a	0
b	13
c	22
f	21
g	10
h	(15)
i	14

Step 5:

a	0
b	13
c	22
f	21
g	10
h	15
i	14

Step 6:

a	0
b	13
c	22
f	21
g	10
h	15
i	14

Step 7:

a	0
b	13
c	22
f	21
g	10
h	15
i	14

b) The distances are a to c: 22
a to f: 21
a to i: 14

Ex 13.1.3 a) We apply the algorithm again:

Step 1:

Vertex	distance
a	0
b	(5)
c	6
f	∞
g	17
h	∞

Step 2:

a	0
b	5
c	(6)
f	∞
g	17
h	12

Step 3:

a	0
b	5
c	6
f	12
g	17
h	12

Step 4:

a	0
b	5
c	6
f	12
g	16
h	12

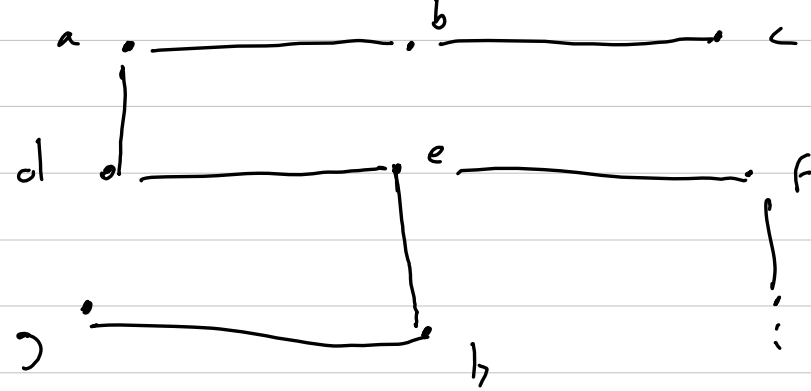
Step 5:

a	0
b	5
c	6
f	12
g	16
h	12

b) a-f : 12 , a-g : 16
a-h : 12

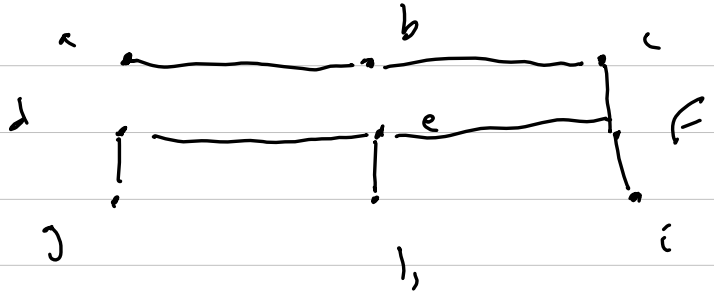
Ex 13.2.1: Kruskal's algorithm:

Add the following edges in order:
1. e-h 2. a-b, b-c, d-e, e-f
3. (Choices made here) a-d
4. (choices) f-i, g-h.

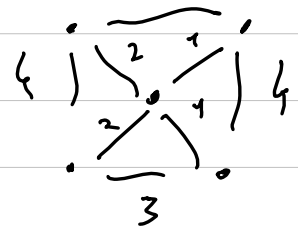


Prim's algorithm:

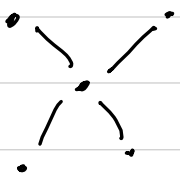
Step 1: a-b, b-c
Step 2: (choices) c-f
Step 3: e-f, e-h, e-d
Step 4: (choices) d-g, f-i



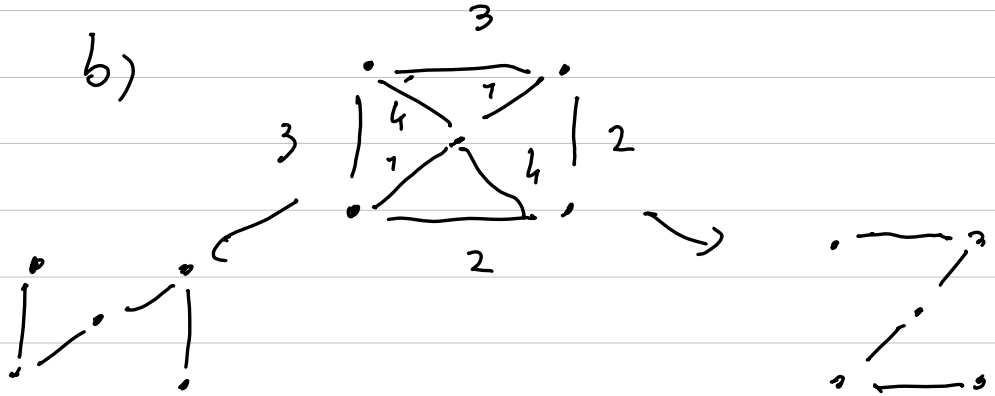
Ex 13.2.2 : a) The following assignment



gives a unique minimal tree:



b)



Ex 12.2.4 : We must connect the cities as follow:

G - S - F - So - B - T - E

(initial letters for cities)