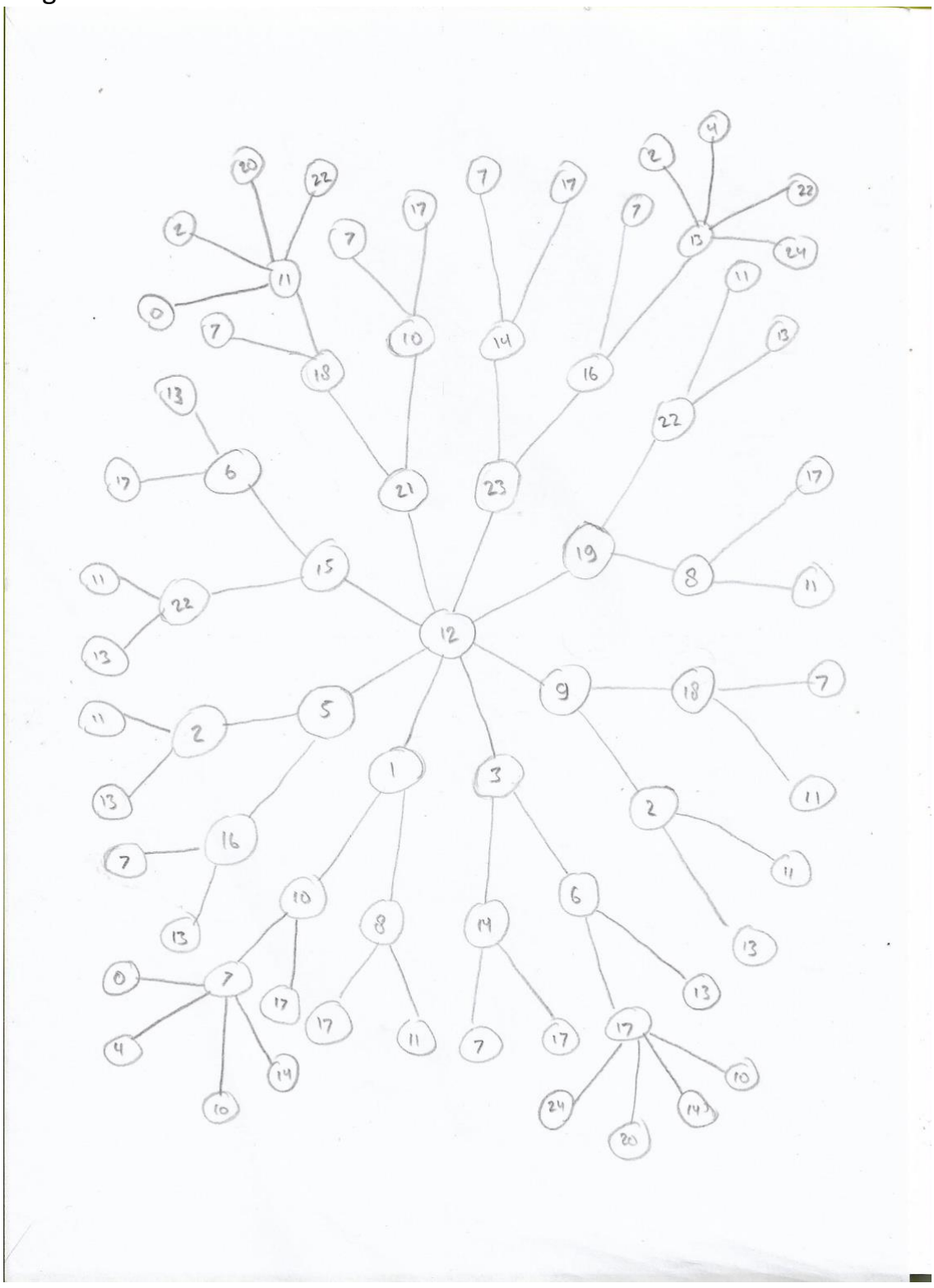
1. Obscure Binary Search Trees

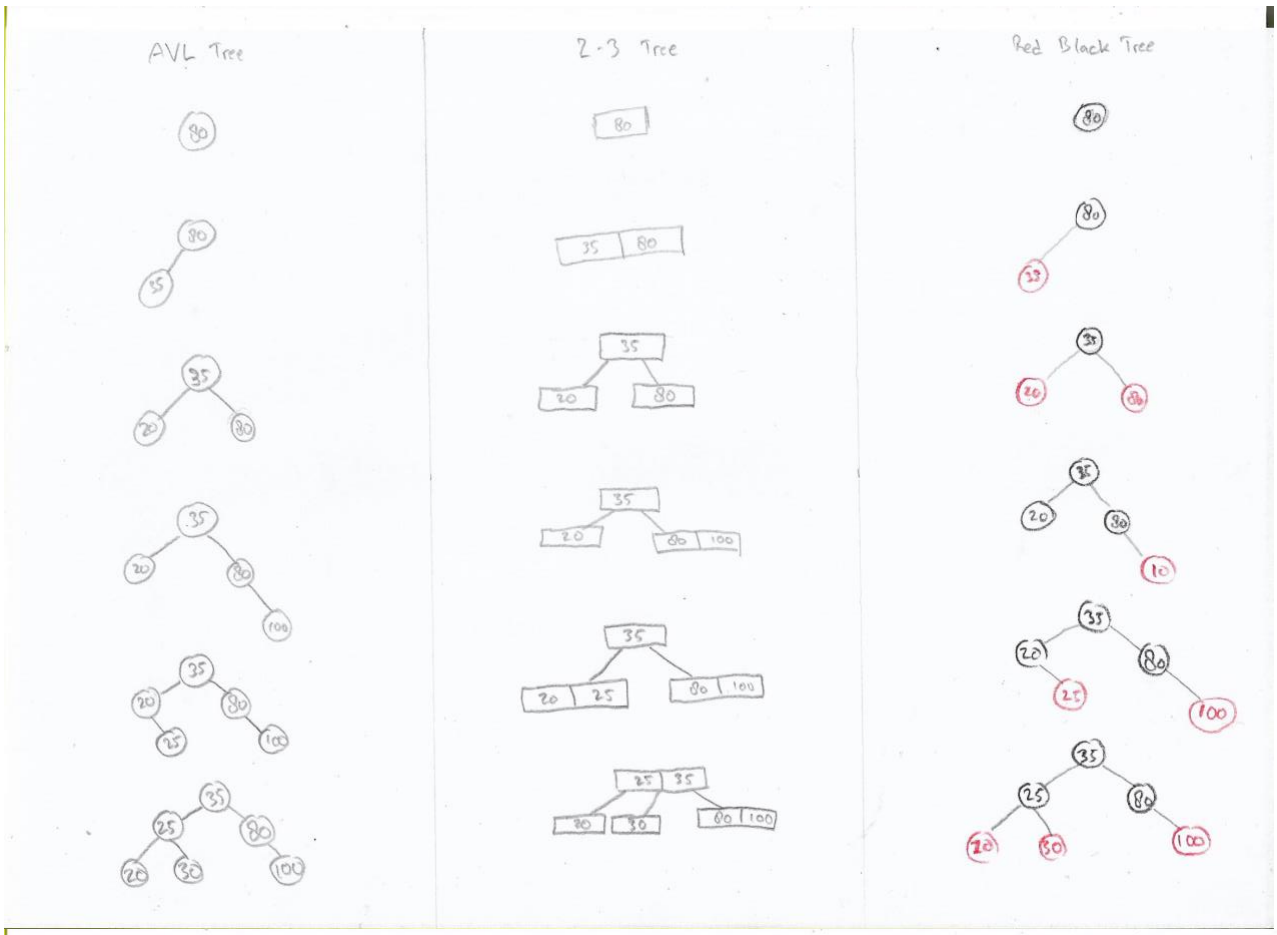
   Splay Tree adalah salah satu alternatif data struktur dari BST dan AVL Tree. Splay Tree ini sama sama binary tree seperti AVL, sama sama menggunakan sistem rotate juga seperti AVL, tetapi perbedaan Spaly Tree dan AVL adalah syarat untuk melakukan rotate pada AVL adalah jika balance factor nya lebih dari 1, sedangkan pada Splay Tree rotate akan dilakukan setiap kali ada searching dan nantinya value yang di search akan berada di paling atas atau base root dari tree. Maka dari hal itu Splay Tree ini akan mengungguli AVL jika dibutuhkan data struktur yang akan sering dilakukan search dengan value yang sama karena tentunya value yang sering di search akan naik dan lebih cepat di search. Selain itu, Splay Tree juga lebih hemat memori disbanding AVL karena tidak perlu menyimpan data balance factor.

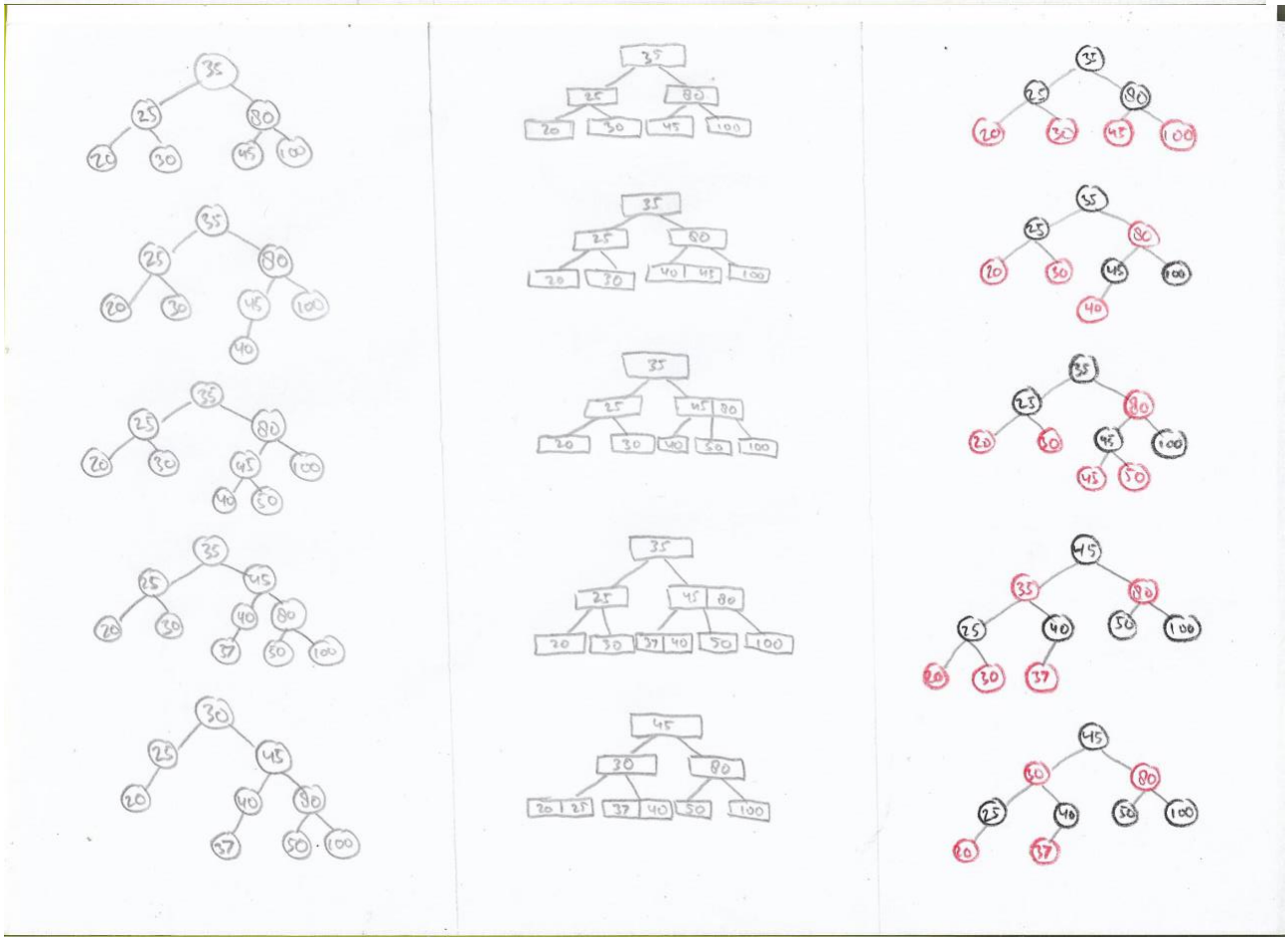
2. Knight Travials

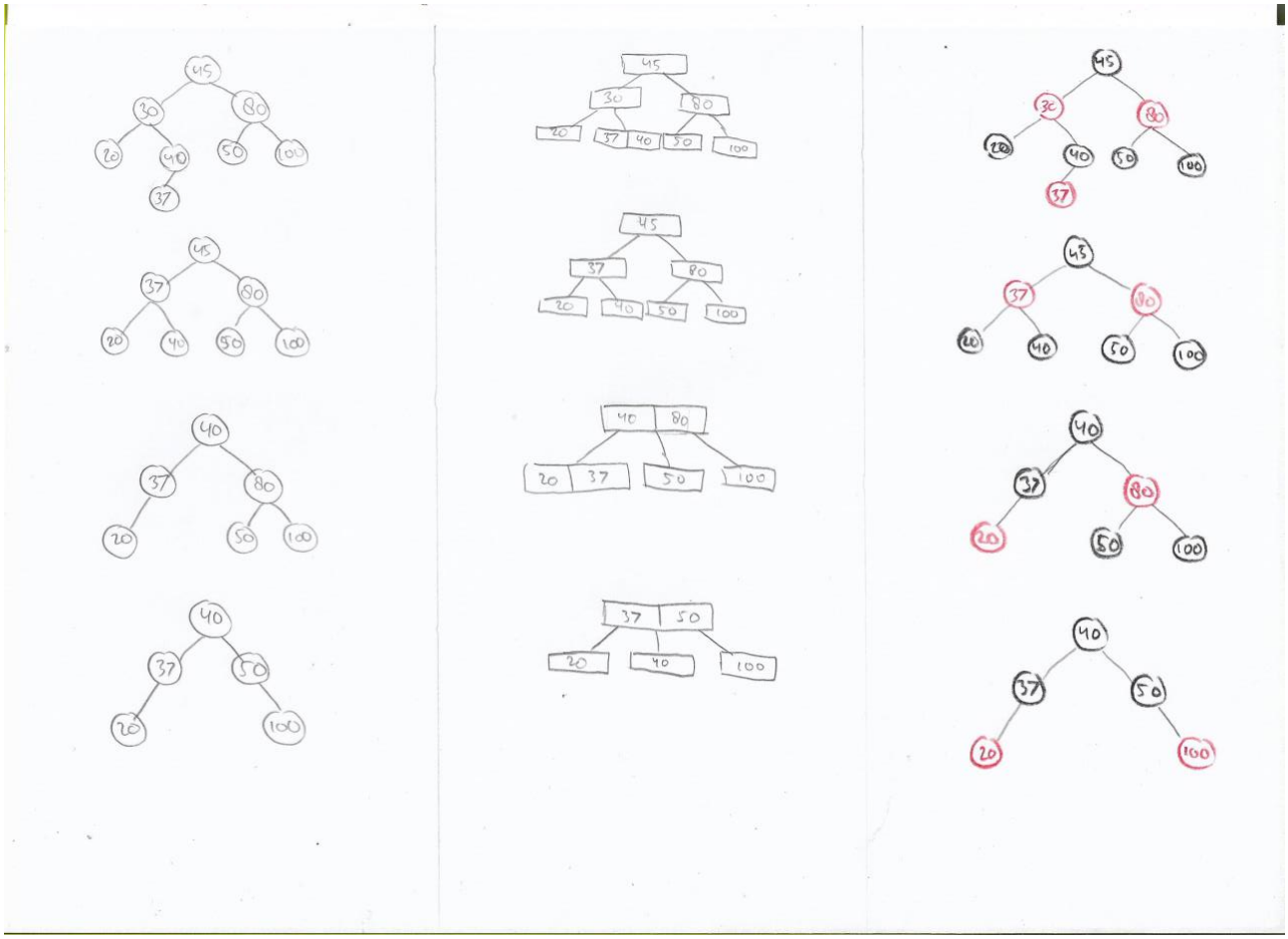


3. Tree Simulations

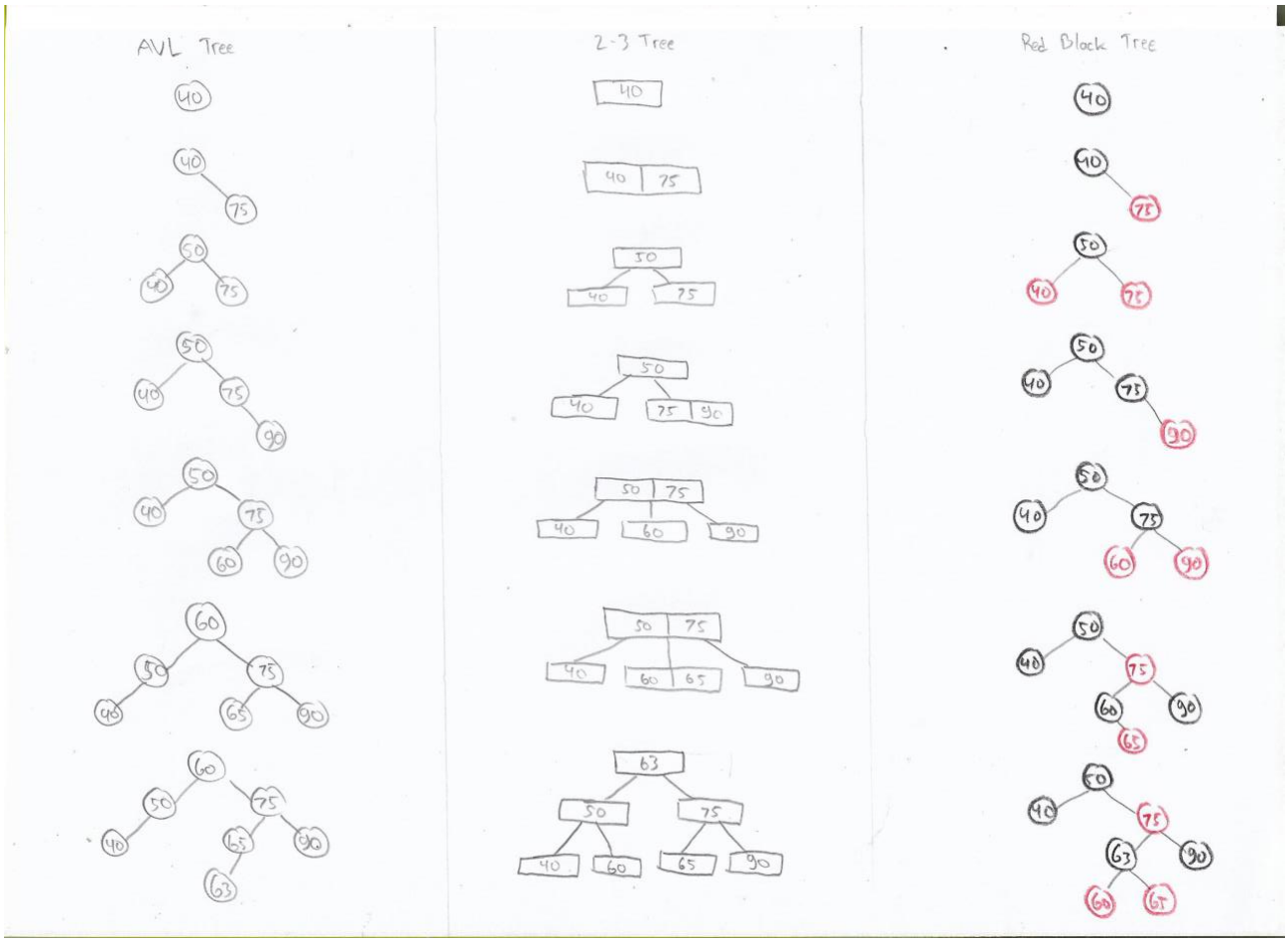
A.

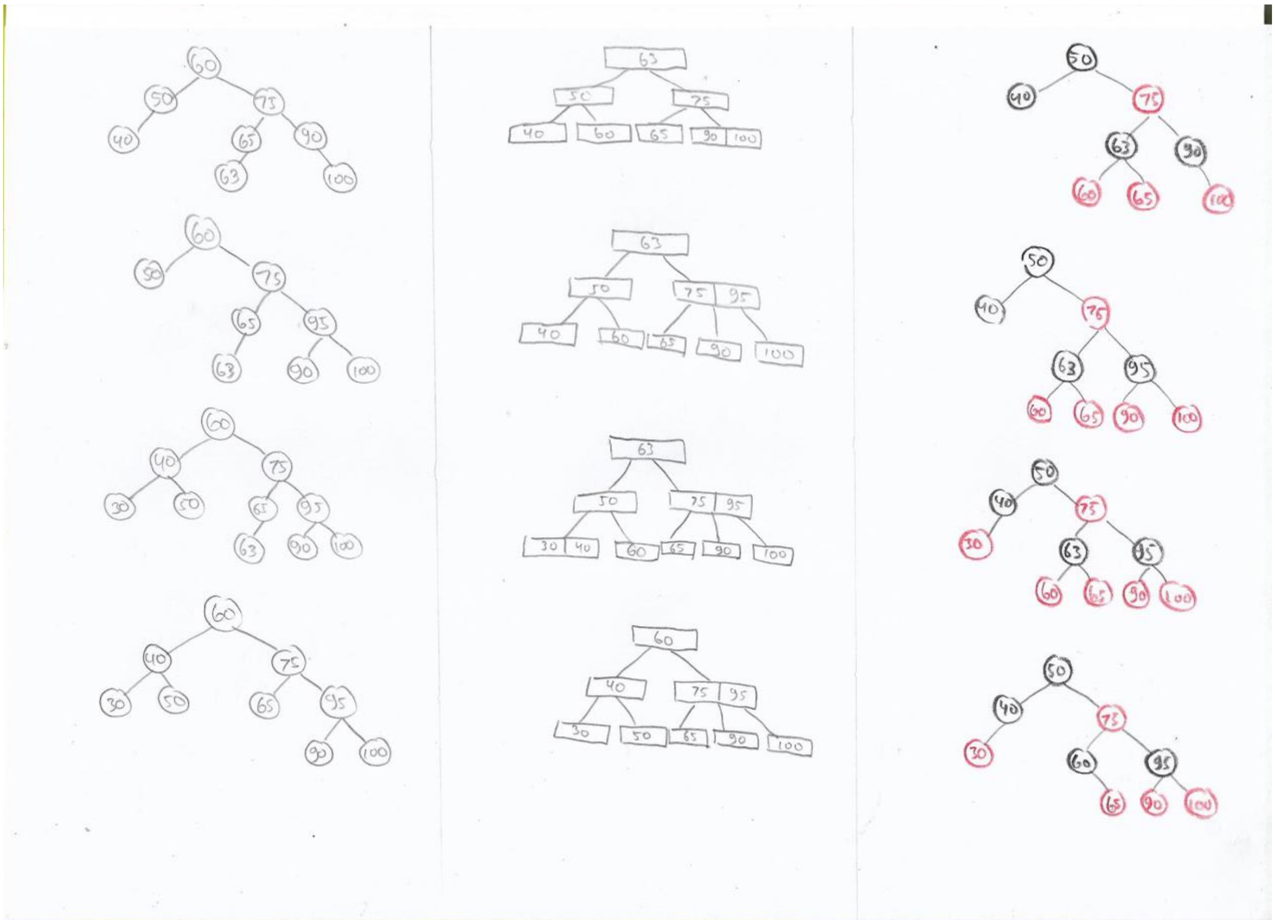


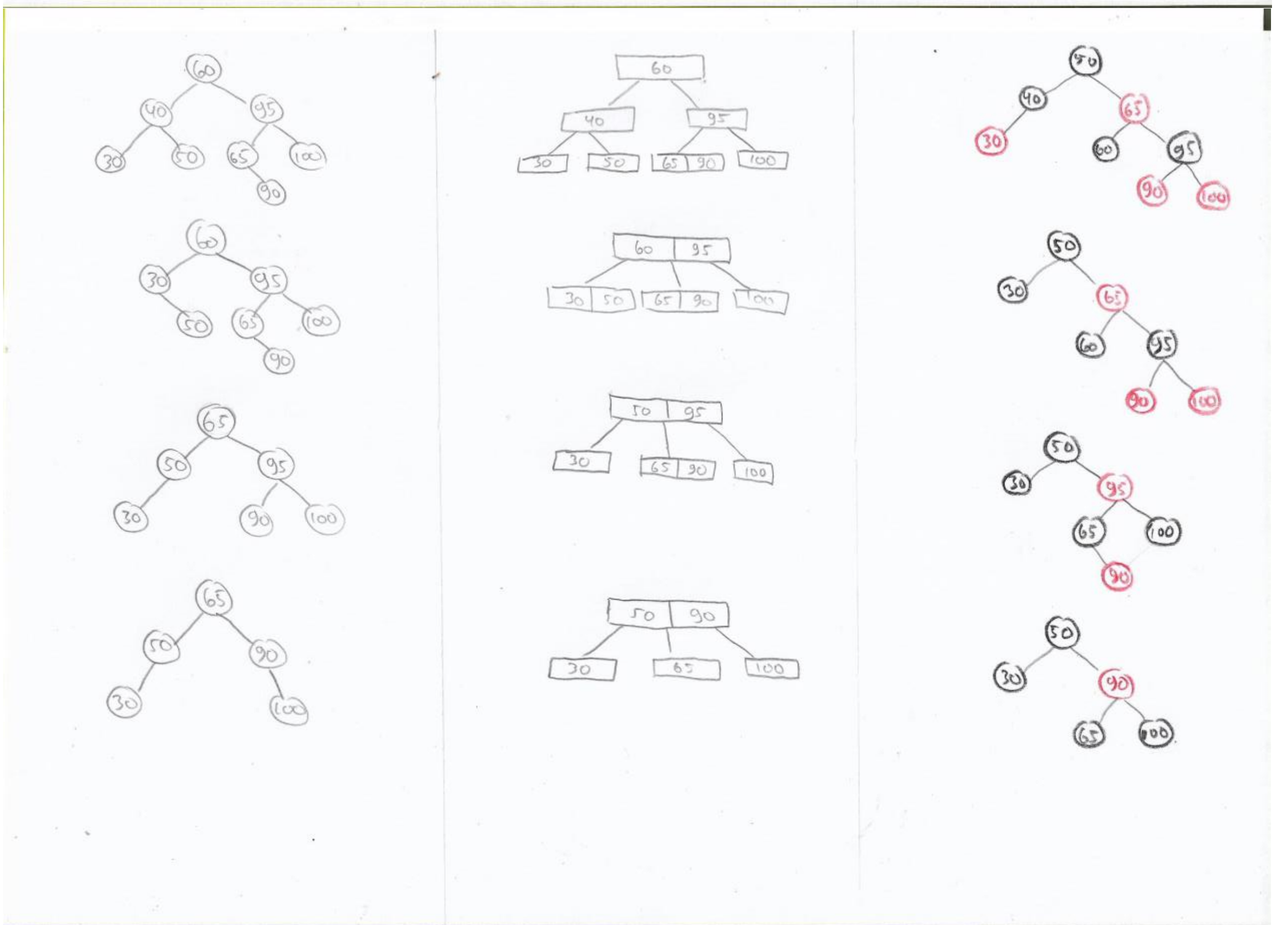



45
30
80
20
40
50
100
37
45
37
80
20
40
50
100
40
37
80
20
50
100
40
37
50
20
100
45
30
80
20
37
40
50
100
45
37
80
20
40
50
100
40
80
20
37
50
100
37
50
20
40
100
45
30
80
20
40
50
100
37
45
37
80
20
40
50
100
40
37
80
20
50
100
40
37
50
20
100

B.

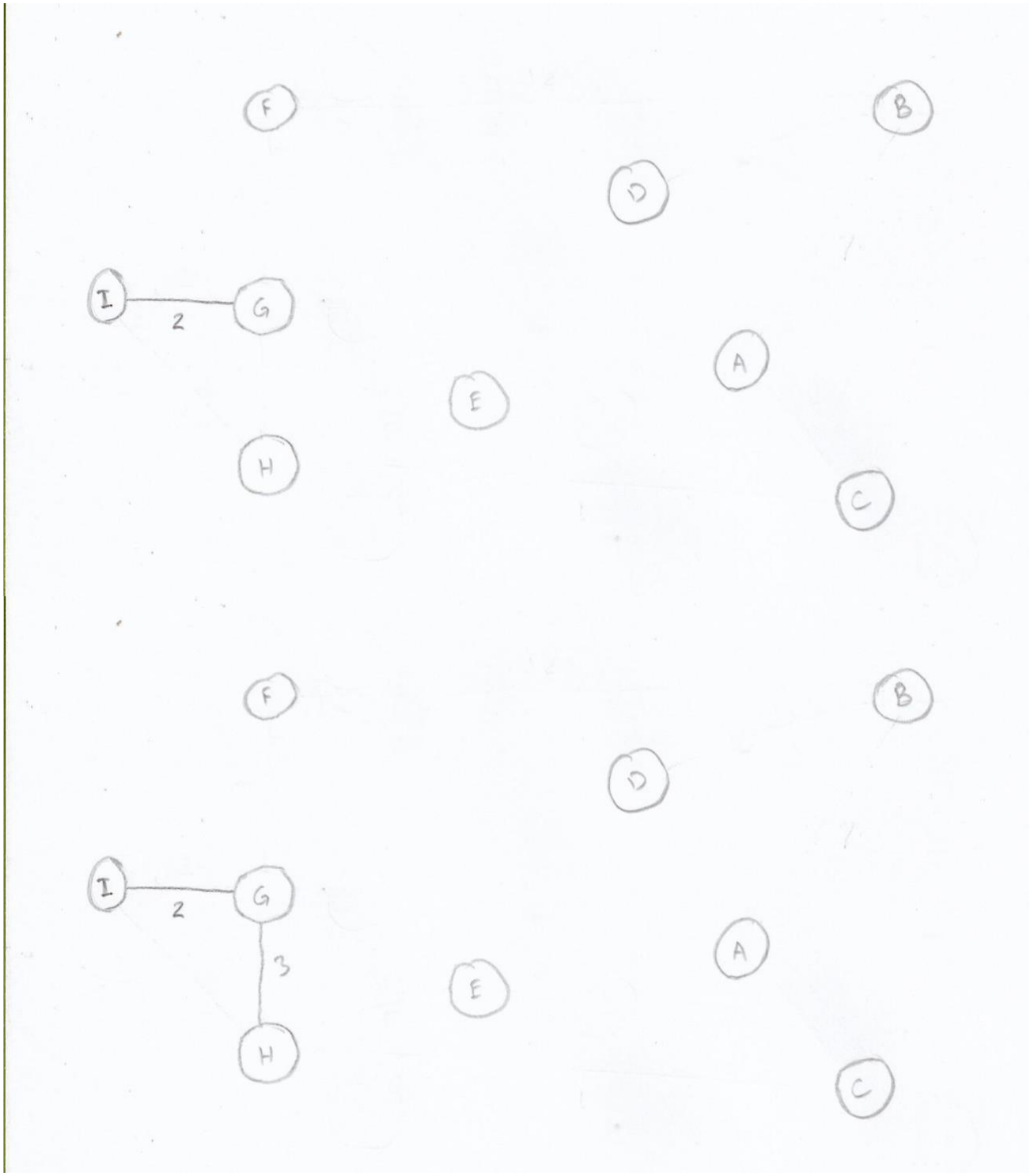

AVL Tree
2-3 Tree
Red Black Tree
40
40
75
50
40
75
50
40
75
90
50
40
75
60
90
60
50
75
40
65
90
60
50
75
40
65
90
63
40
60
40
75
50
40
75
50
40
75
90
50
40
75
60
90
50
40
75
60
90
65
50
40
75
63
90
60
65

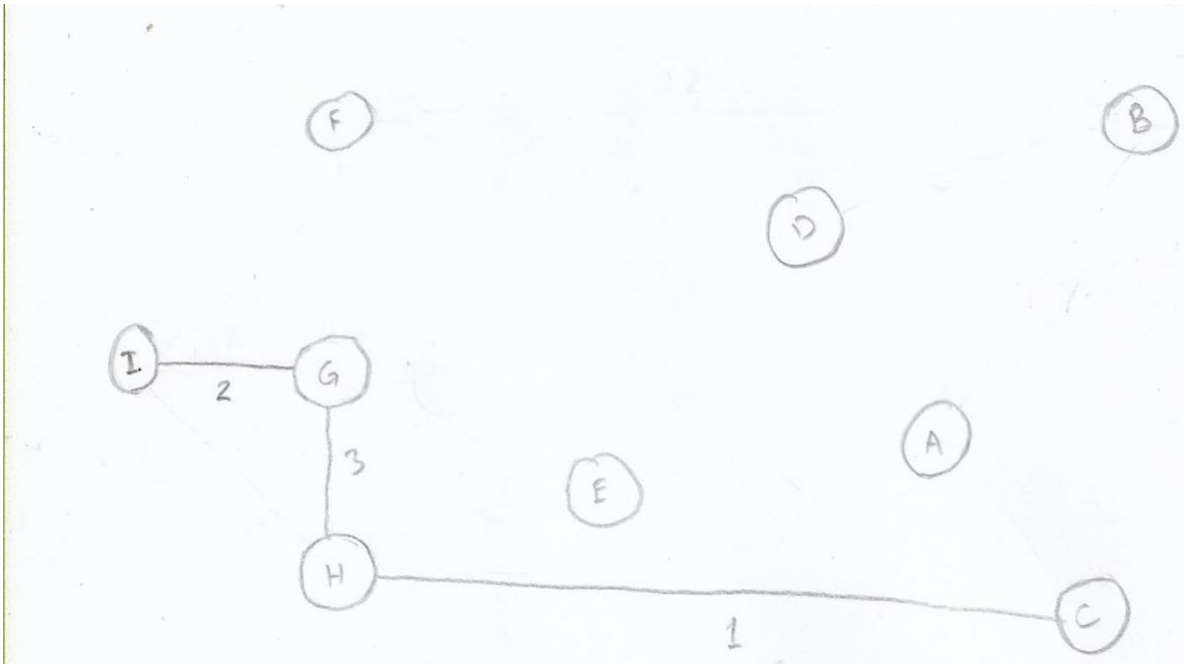

60 50 75 40 65 90 63 100
60 50 75 65 95 63 90 100
60 40 75 30 50 65 95 63 90 100
60 40 75 30 50 65 95 90 100
63 50 75 40 60 65 90 100
63 50 75 95 40 60 65 90 100
63 50 75 95 30 40 60 65 90 100
60 40 75 95 30 50 65 90 100
50 40 75 63 90 60 65 100
50 40 75 63 95 60 65 90 100
50 40 75 30 63 95 60 65 90 100
50 40 75 30 60 95 65 90 100

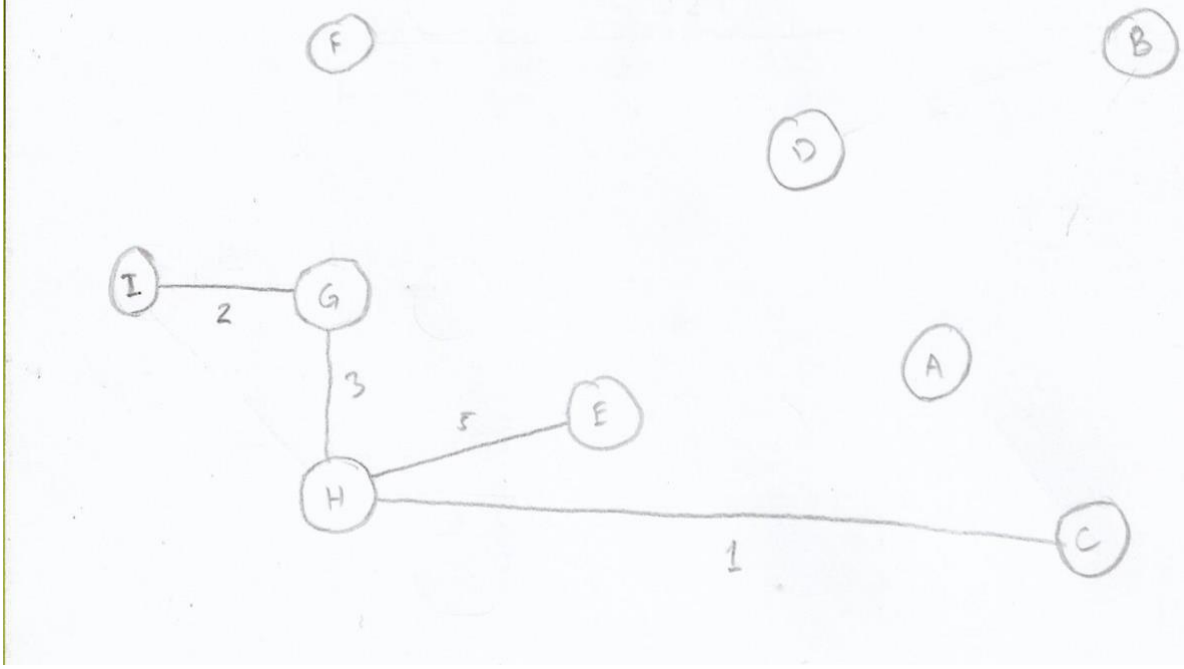

60 40 95 30 50 65 100 90
60 30 95 50 65 100 90
65 50 95 30 90 100
65 50 90 30 100
60 40 95 30 50 65 90 100
60 95 30 50 65 90 100
50 95 30 65 90 100
50 90 30 65 100
50 40 65 30 60 95 90 100
50 30 65 60 95 90 100
50 30 95 65 100 90
50 30 90 65 100

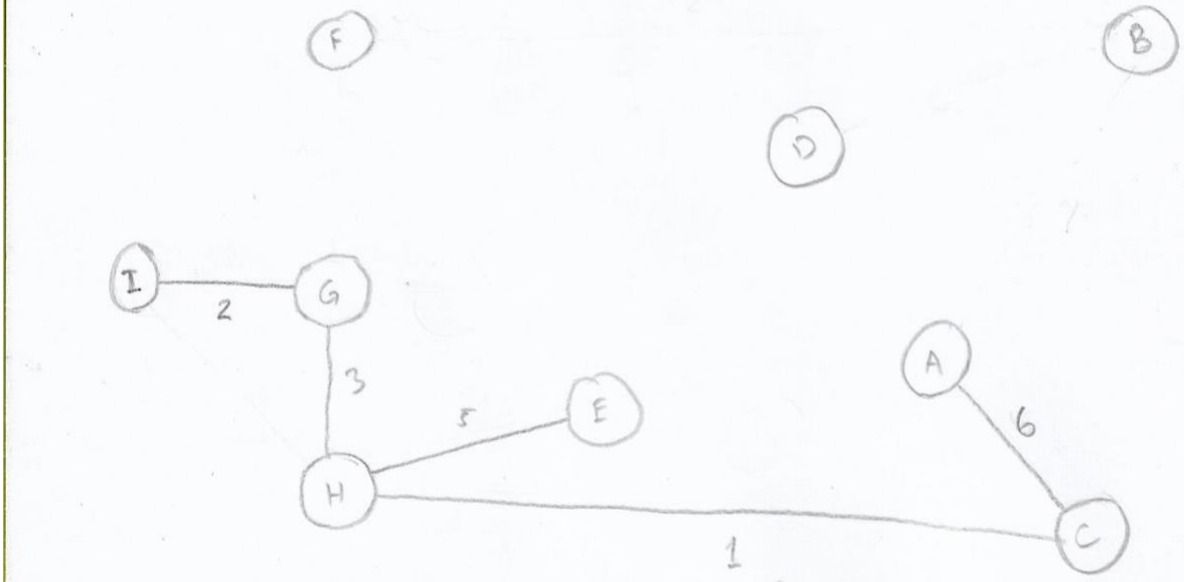
4. Disjoint Set & Graph

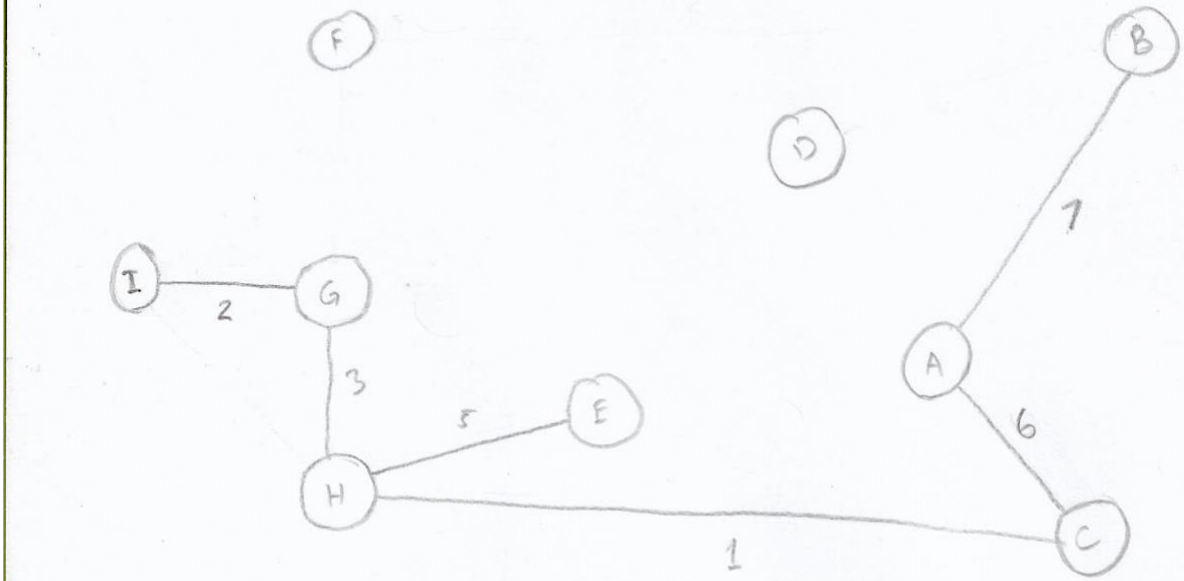
Prim

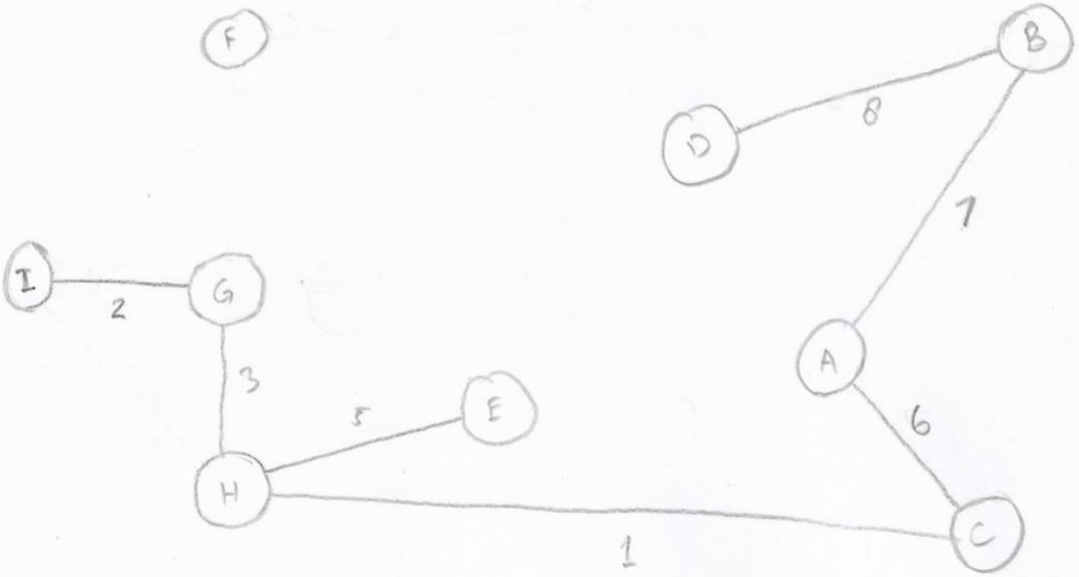


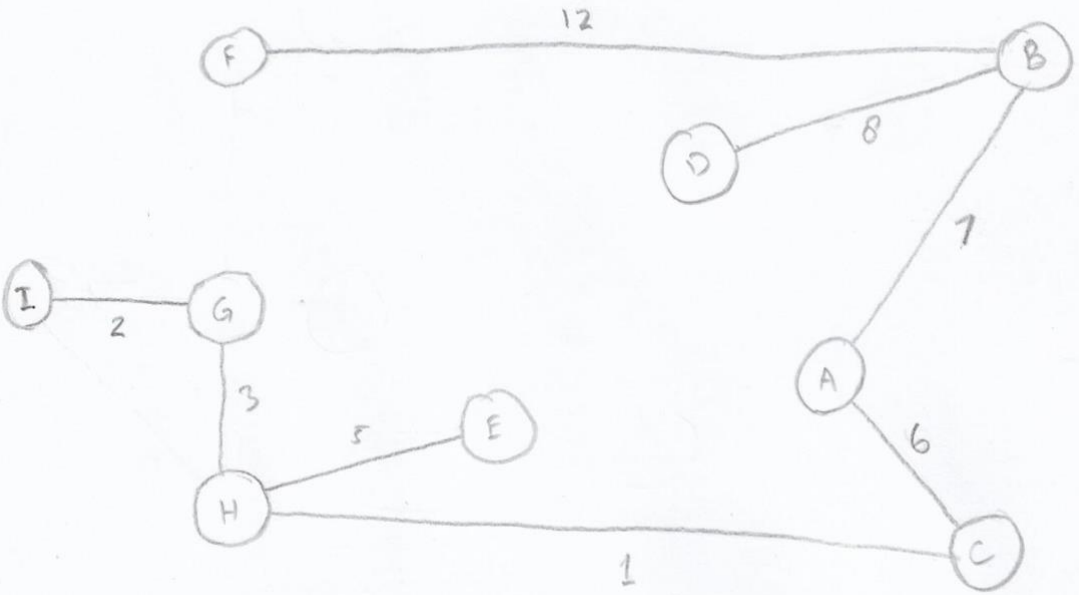

F
B
D
I
2
G
3
A
E
H
1
C


F
B
D
I
2
G
3
A
5
E
H
1
C

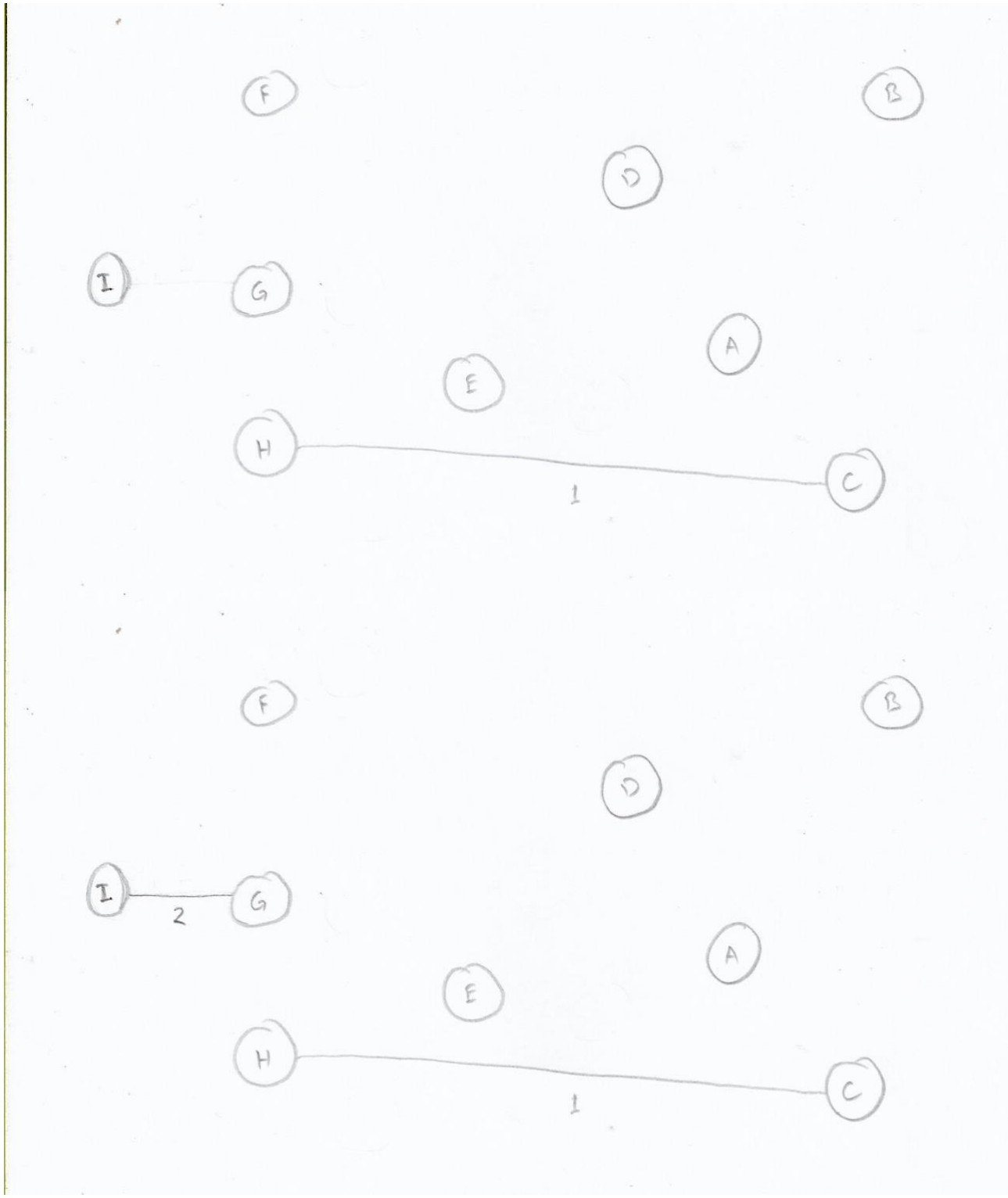

F
B
D
I
G
2
3
5
E
A
6
H
C
1

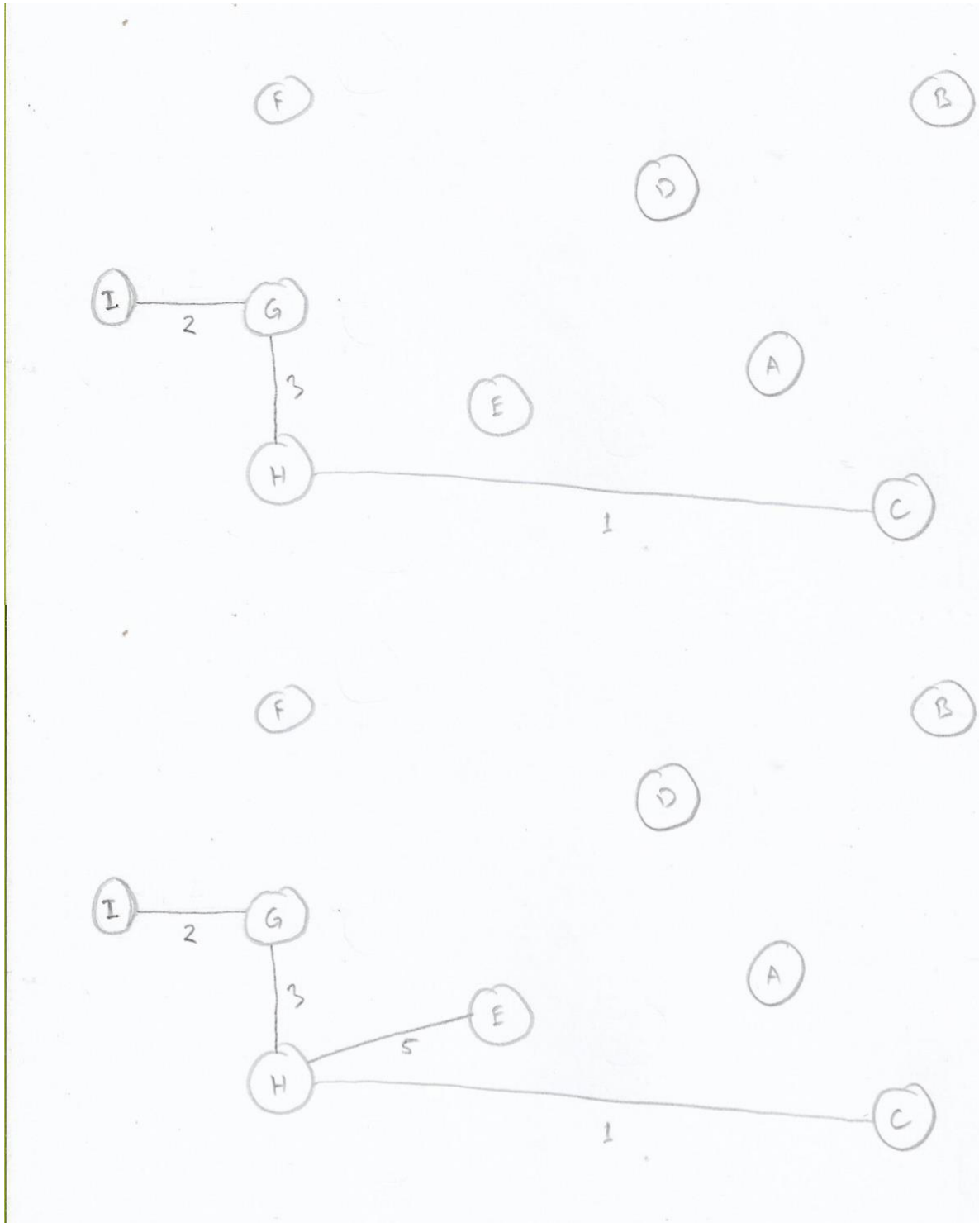

F
B
D
1
I
G
2
3
A
5
E
6
H
C
1


F
B
D
8
7
I
2
G
A
3
5
E
6
H
C
1

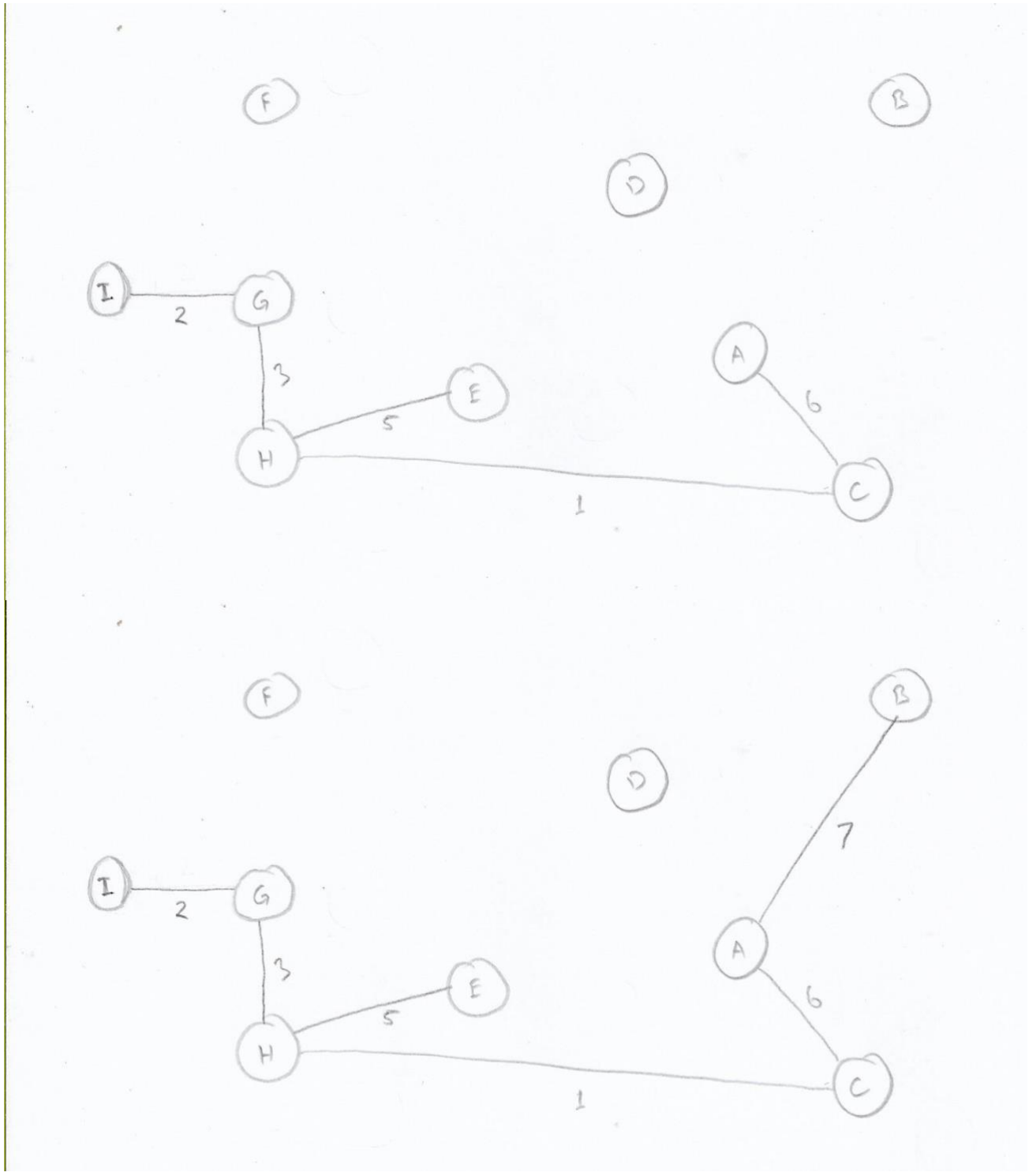

12
F
B
D
8
7
I
2
G
A
3
5
E
6
H
C
1

Kruskal

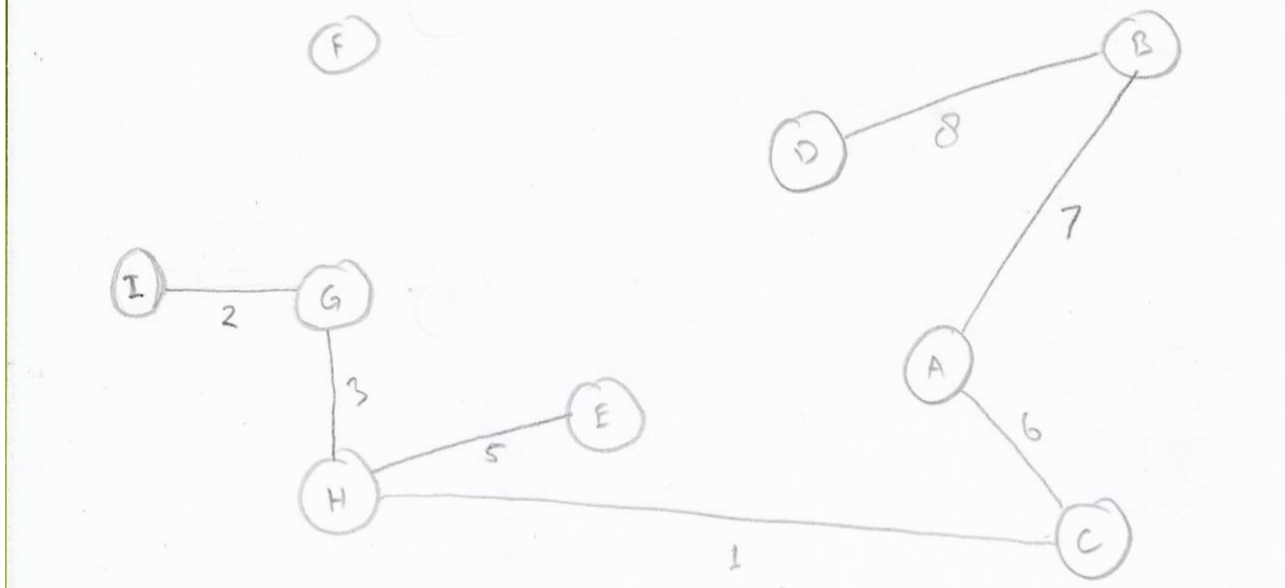


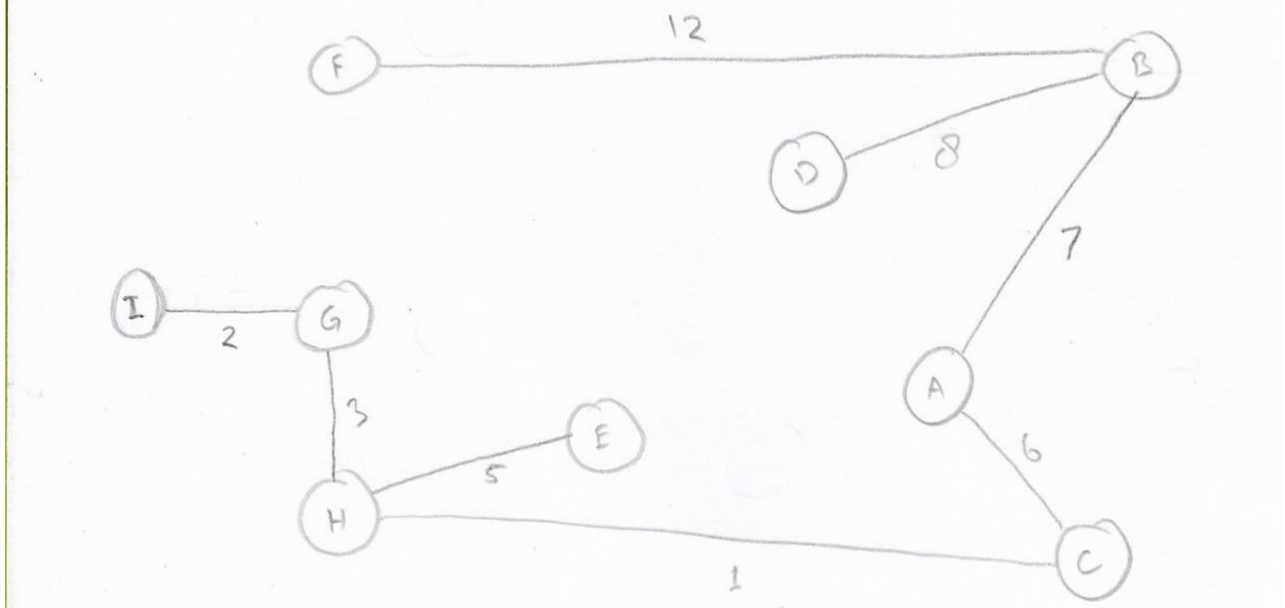

F
B
D
I
2
G
3
A
E
H
1
C

F
B
D
I
2
G
3
A
E
5
H
1
C

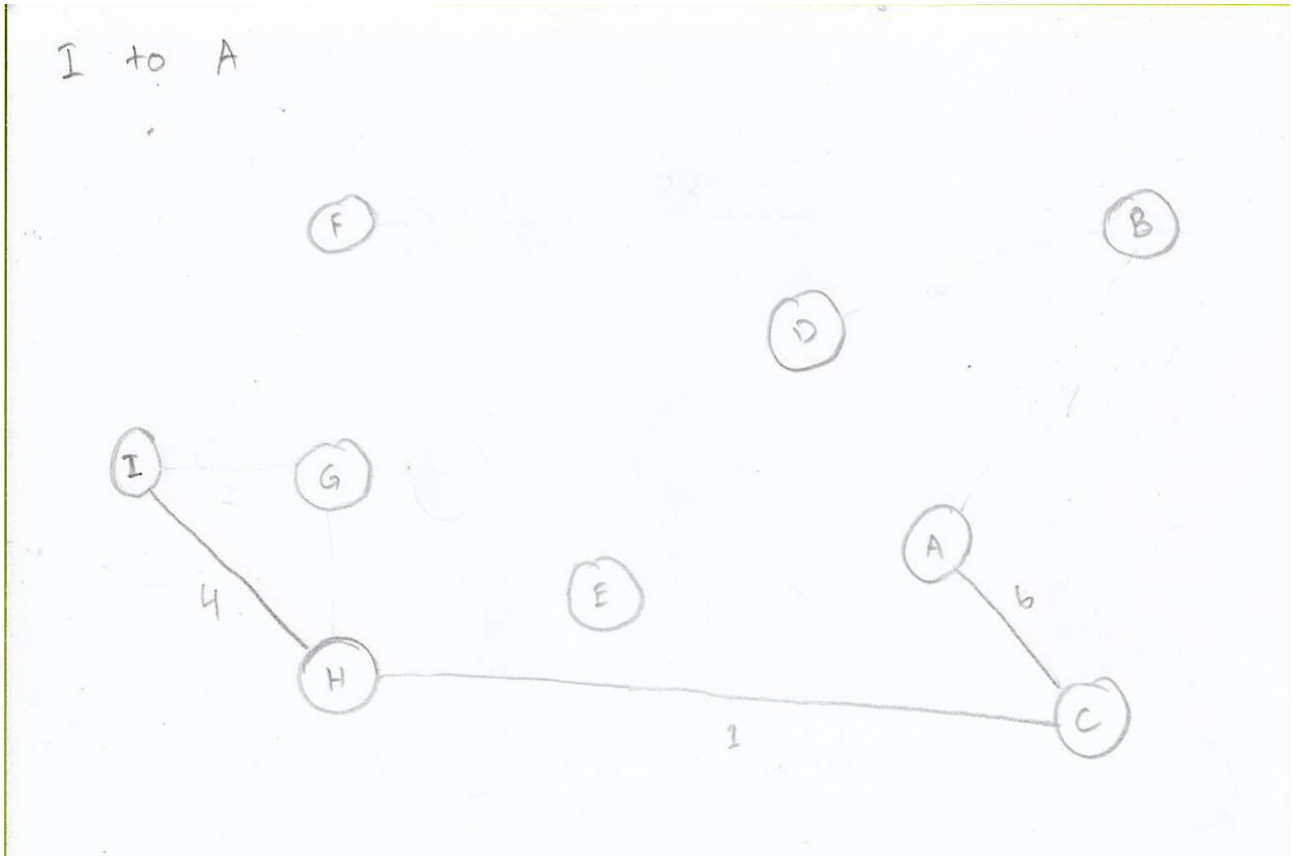

F
B
D
I
G
2
3
E
5
A
6
H
C
1

F
B
D
7
I
G
2
3
A
E
6
5
H
C
1


F
B
D
8
7
I
2
G
A
3
E
6
5
H
C
1

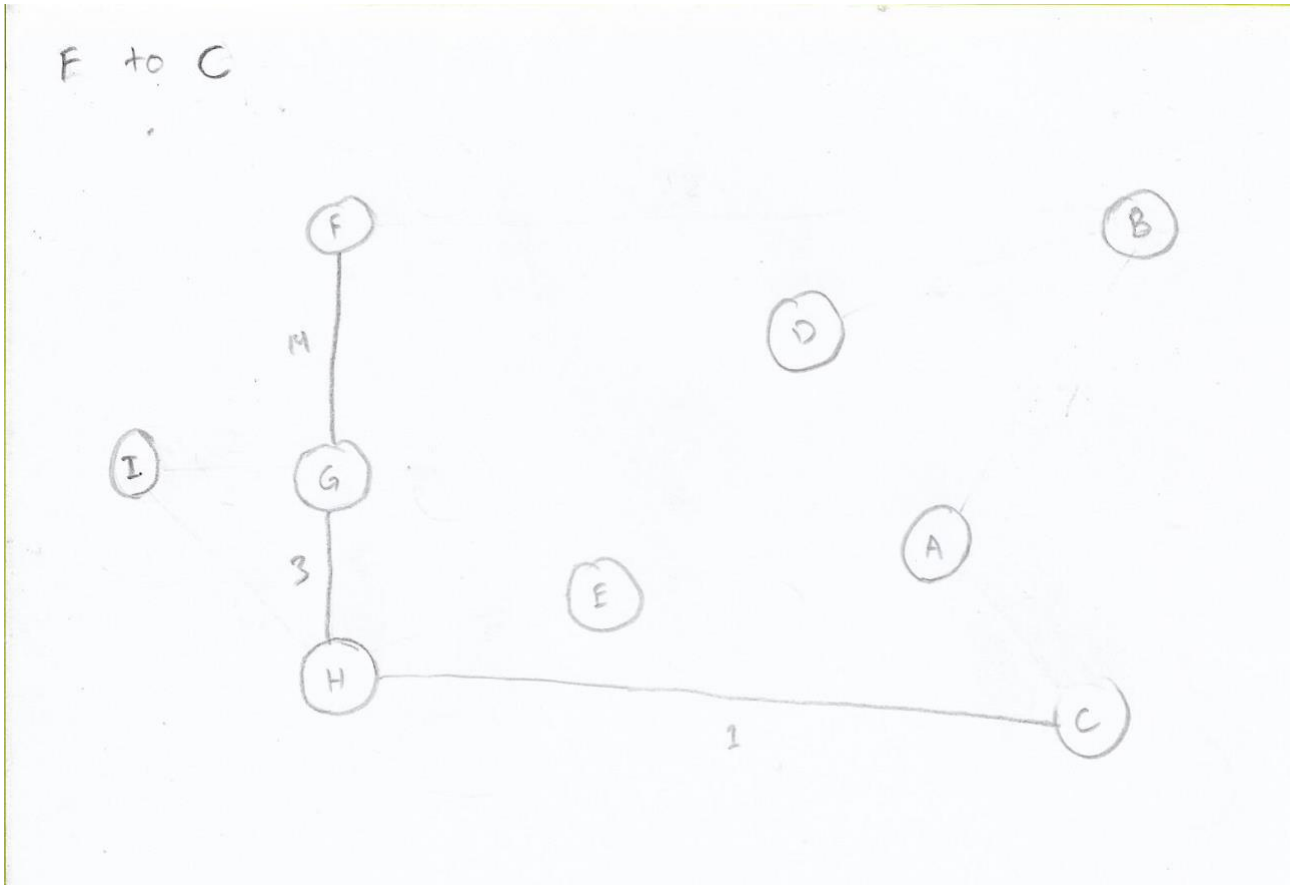

12
F
B
D
8
7
I
2
G
A
3
E
6
5
H
C
1

Shortest Path I to A



I -> H -> C -> A
Cost = 11

Shortest Path F to C



F -> G -> H -> C
Cost = 18