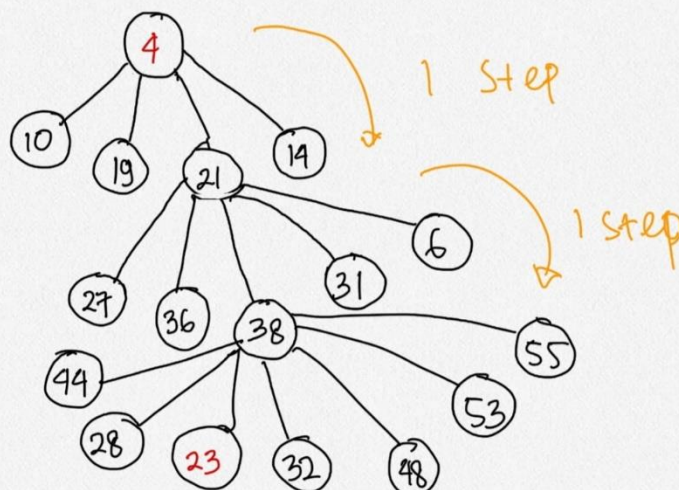


1. Splay Tree merupakan salah satu algoritma balancing sebuah Binary Tree. Untuk melakukan balancing tersebut, tree ini menerapkan metode zig zag. Pada tree tersebut juga ada sebuah aturan dimana node yang di-insert akan menjadi root dari tree tersebut dan node – node dibawahnya tentu diseimbangkan berdasarkan balance factor layaknya seperti AVL Tree. Pada tree ini, dapat dikatakan lebih efektif dibandingkan Binary Search Tree (BST) karena pada Splay tree, worst case dari search, insert, dan delete adalah  $O(\log(n))$  dimana  $\log(n)$  merupakan log basis 2.
2. Menggunakan metode BFS.

|    |    |    |    |    |    |    |    |
|----|----|----|----|----|----|----|----|
| 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  |
| 9  | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
| 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 |
| 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 |
| 33 | 34 | 35 | 36 | 37 | 38 | 39 | 40 |
| 41 | 42 | 43 | 44 | 45 | 46 | 47 | 48 |
| 49 | 50 | 51 | 52 | 53 | 54 | 55 | 56 |
| 57 | 58 | 59 | 60 | 61 | 62 | 63 | 64 |

Find step from 4 to 23



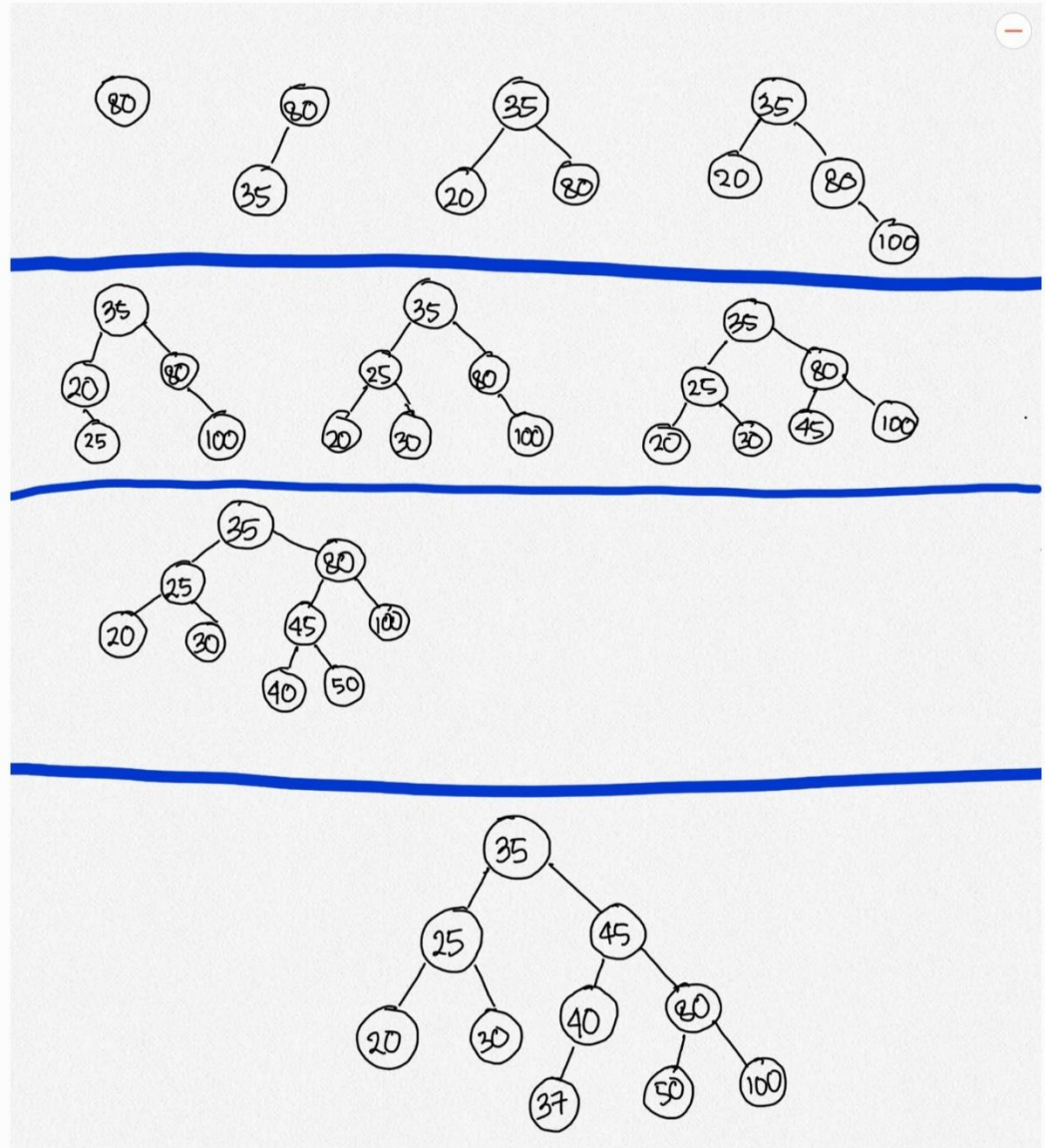
Total = 2 step

3. –

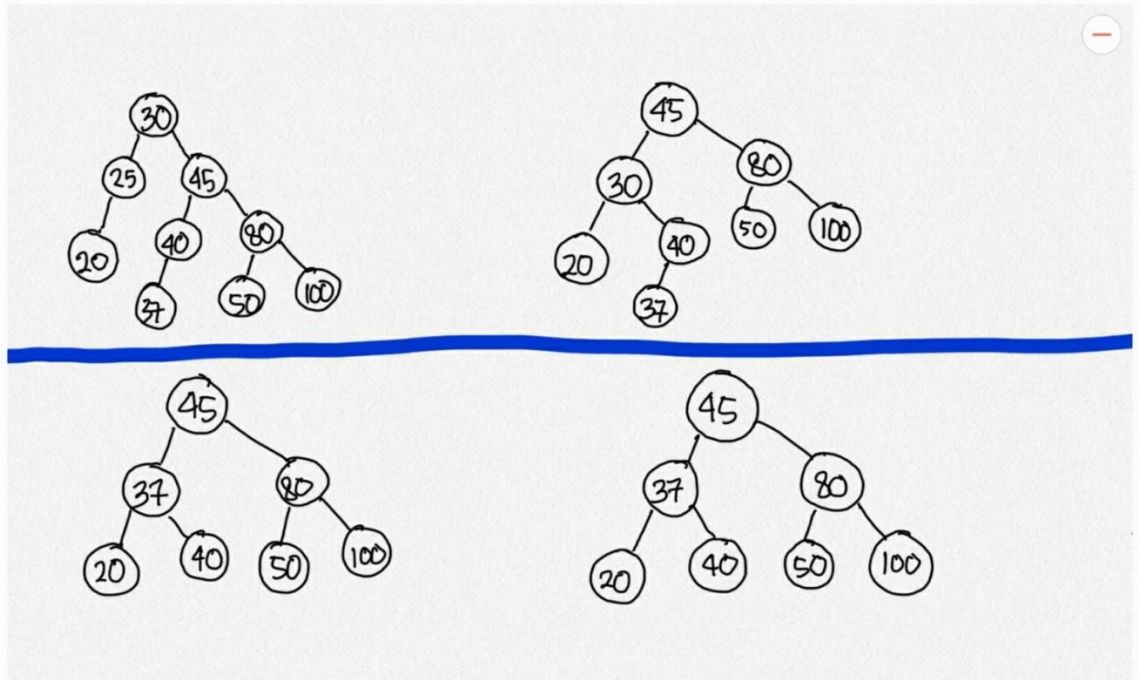
4. Tree  
a.

## AVL TREE

INSERT

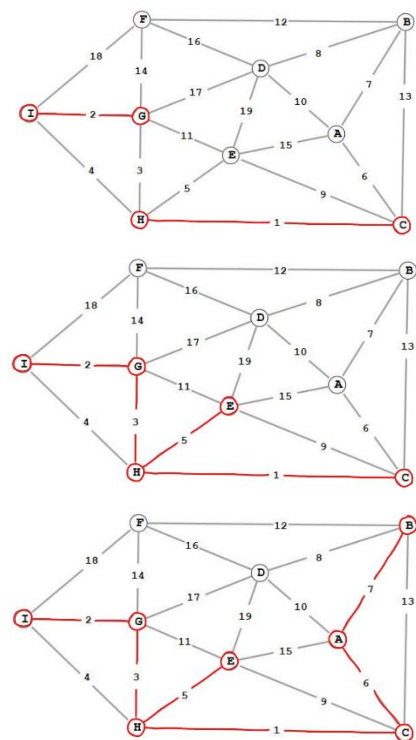
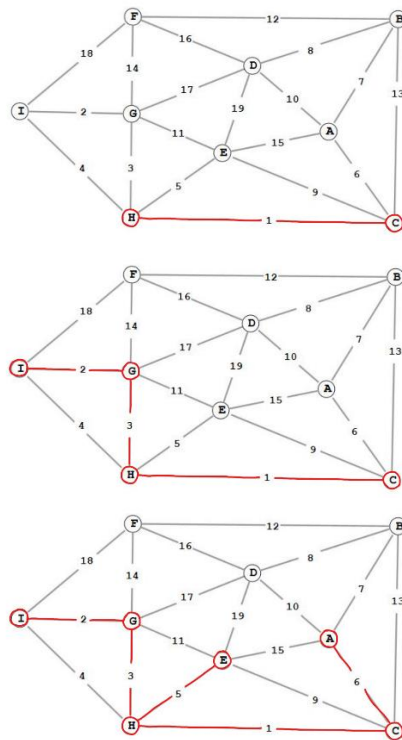


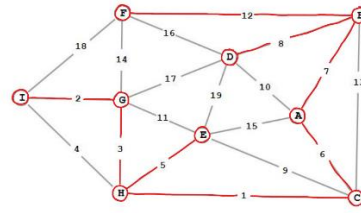
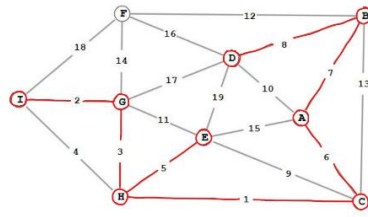
Delete



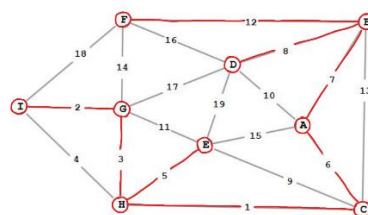
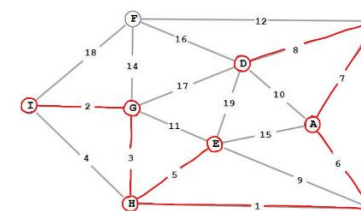
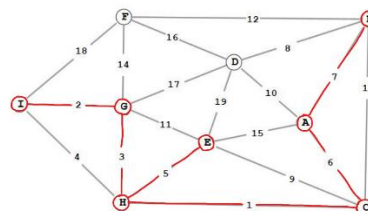
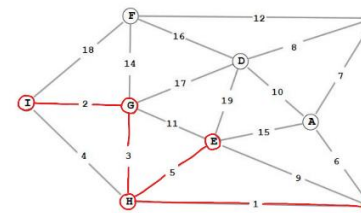
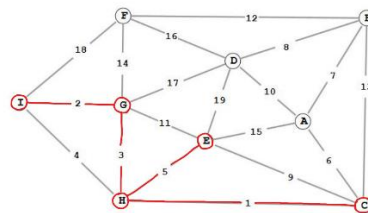
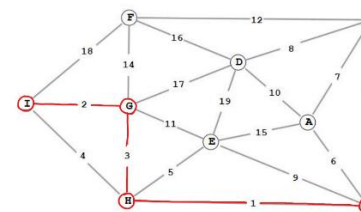
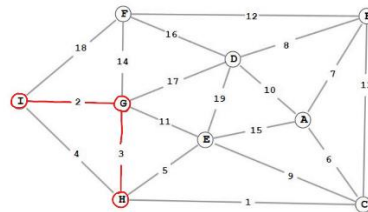
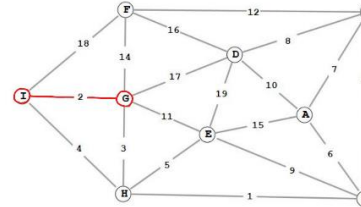
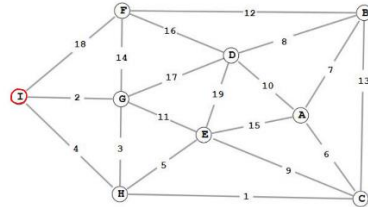
## 5. Minimum Spanning TREE

a. Kruskal (urutan dari kiri ke kanan)



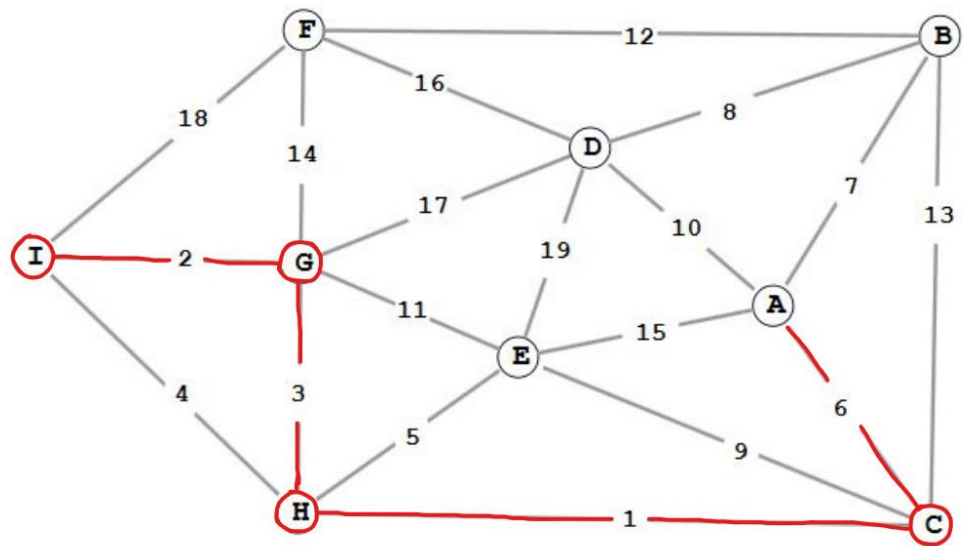


b. Prim (urutan dari kiri ke kanan)





Dijkstra  
I to A



F to C

