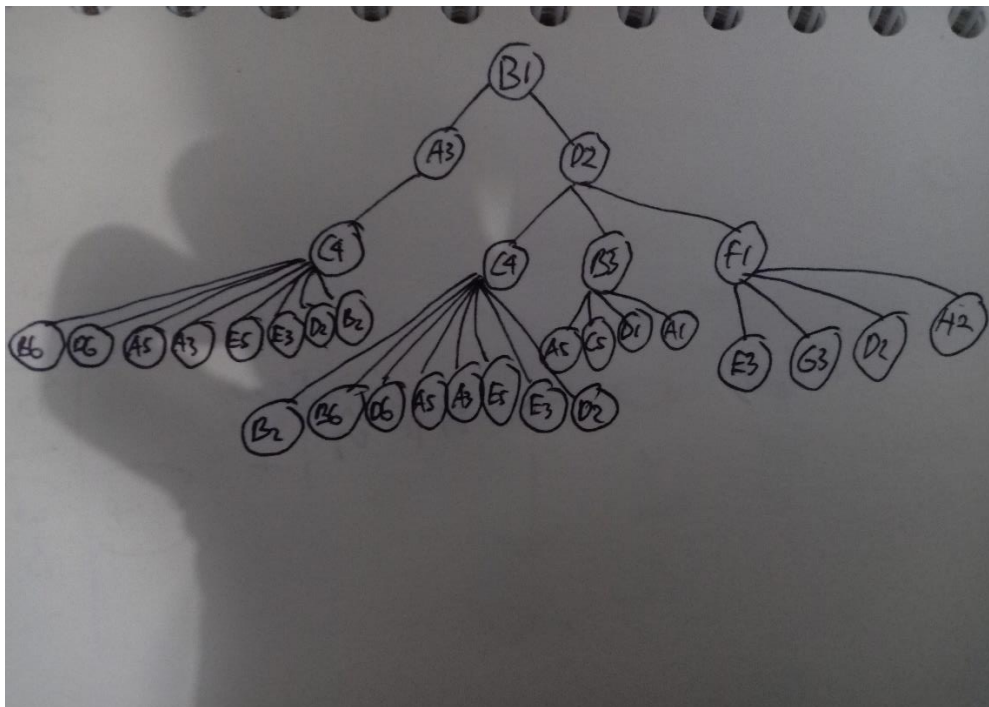


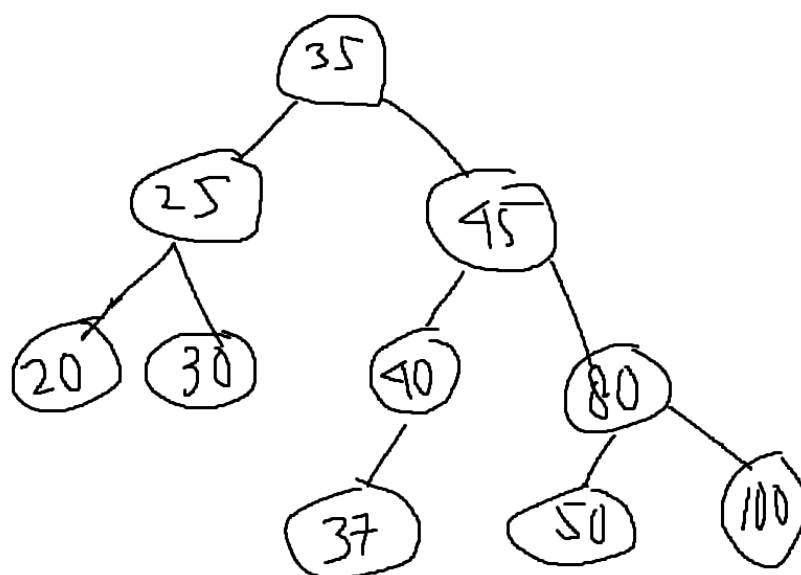
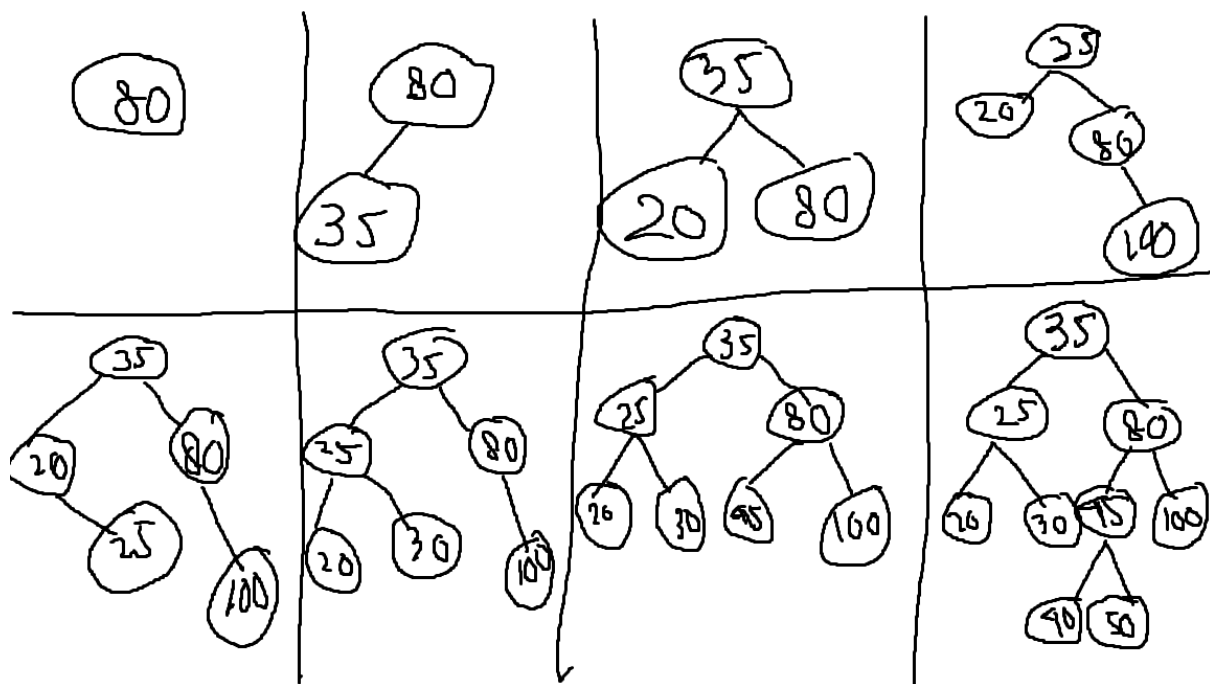
Clevert Ganda Saputra (CheeTos)

Final Project Bootcamp

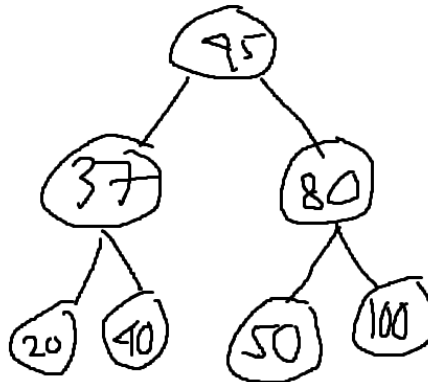
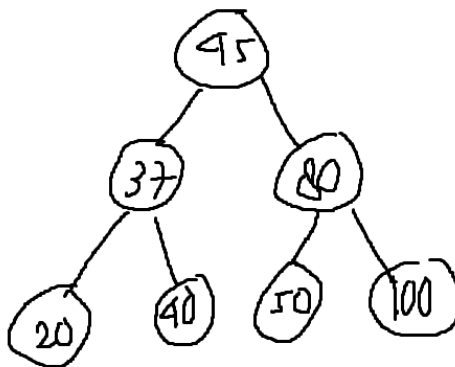
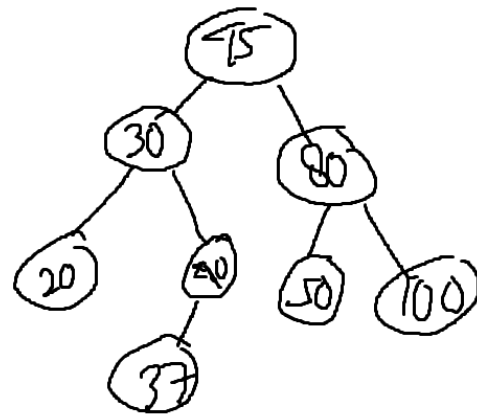
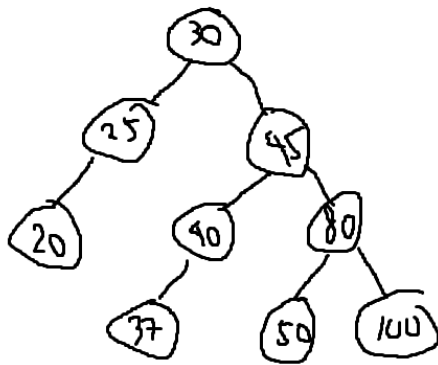
1. AVL Tree adalah sebuah algoritma BST yang menurut saya sangat bagus, syarat AVL Tree disini adalah bahwa antara subtree kiri dan subtree kanan memiliki perbedaan level maksimal 1, AVL Tree ini digunakan untuk menyeimbangkan sebuah BST. Waktu pencarian menjadi lebih sederhana dan lebih singkat dengan AVL tree ini, yang dilakukan oleh AVL Tree ini antara lain single rotation dan double rotation. Di AVL Tree ini lebih efektif daripada BST jika dilihat dari worst casenya
2. –Contoh Tree



3. Dengan Gap buffer, Gap Buffer adalah struktur data yang digunakan untuk mengedit dan menyimpan teks secara efisien yang sedang diedit. Ini juga mirip dengan larik tetapi celah dimasukkan dalam larik untuk menangani banyak perubahan pada kursor. Fungsionalitasnya antara lain : insert, left, right dan grow.
4. AVL Tree (maap ko ga rapi)
 - a) Insert

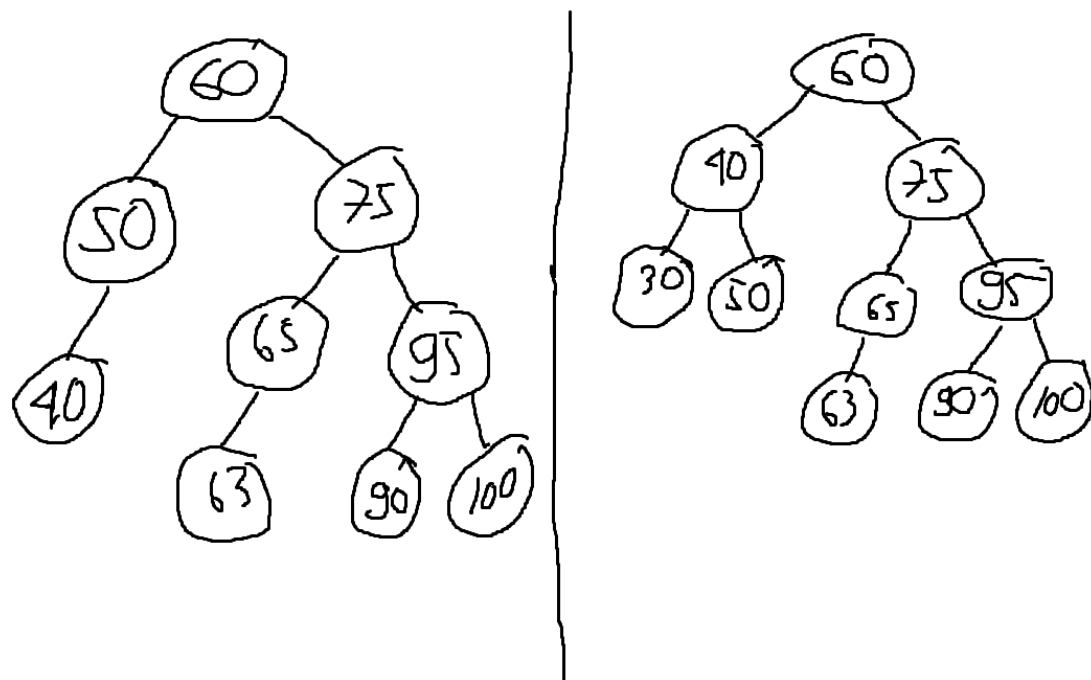
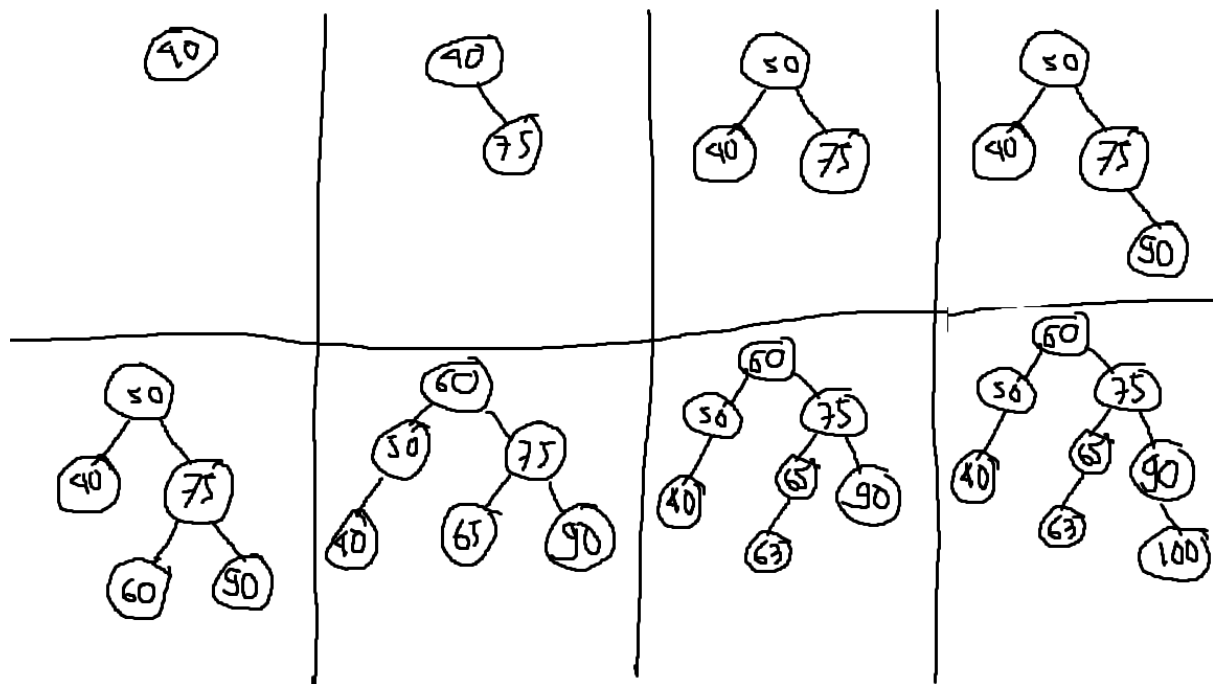


Delete

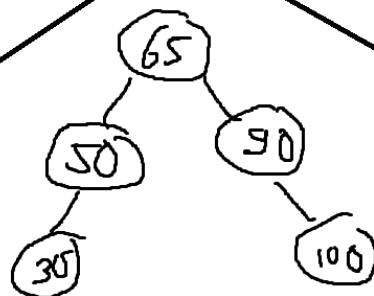
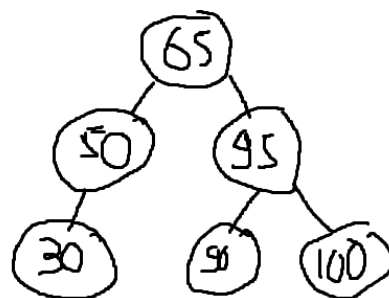
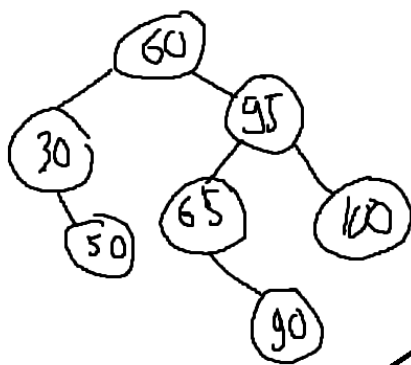
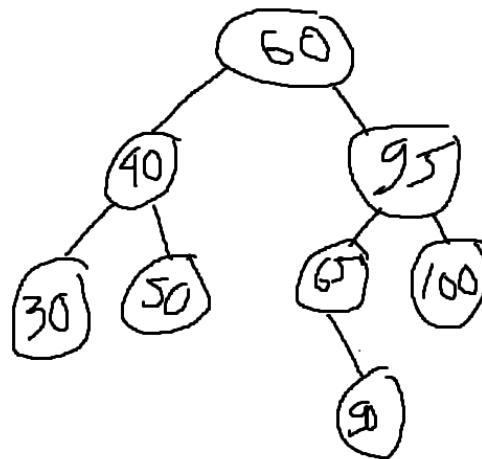
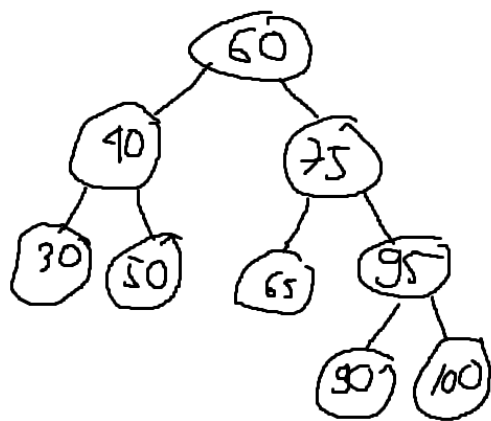


b)

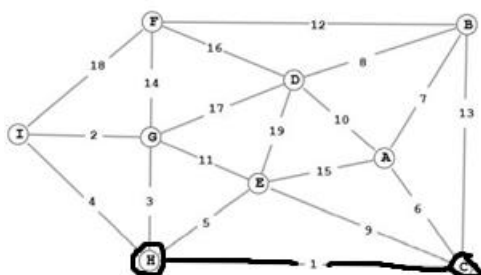
insert

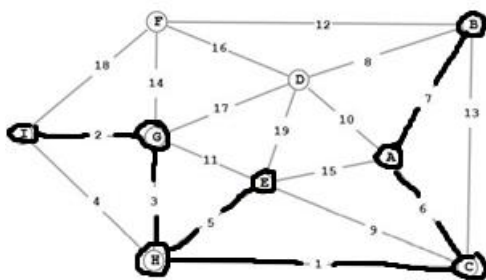
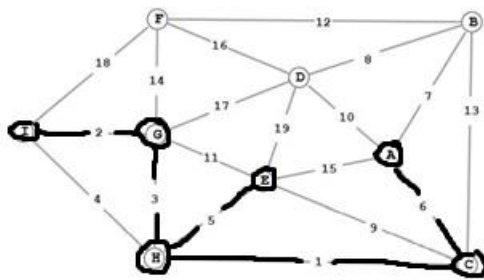
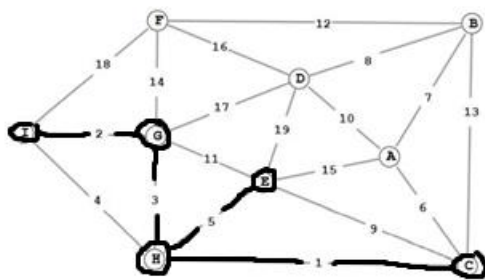
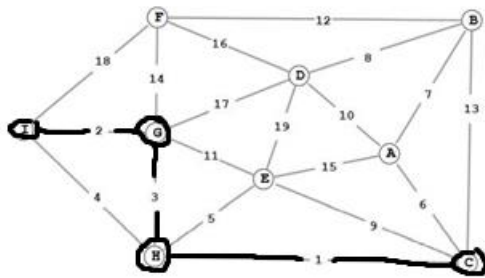
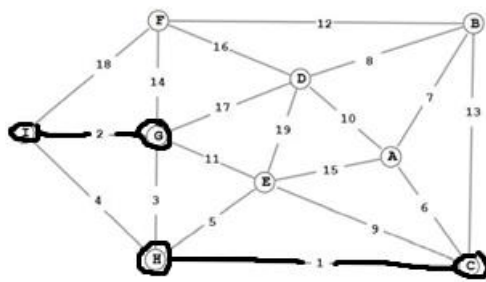


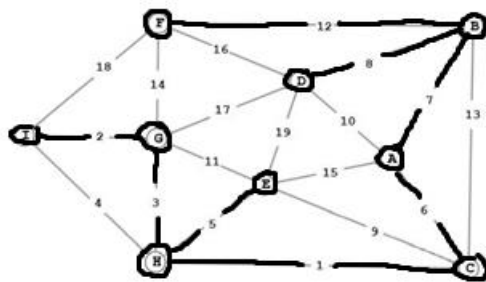
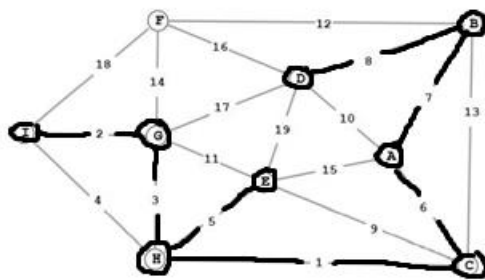
Delete



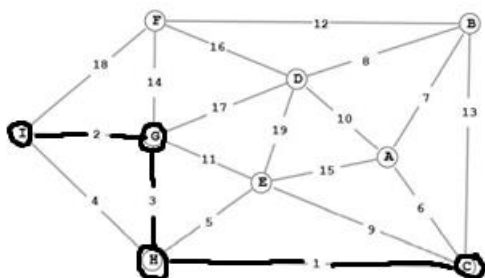
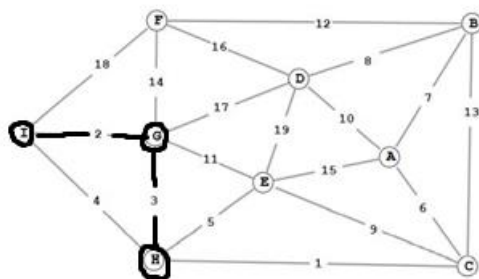
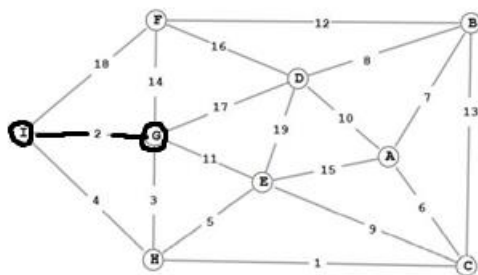
5. Minimum Spanning Tree
a. Kruskal(kiri->kanan)

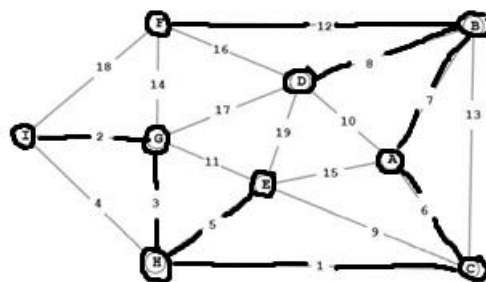
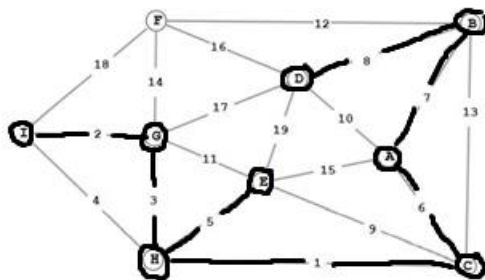
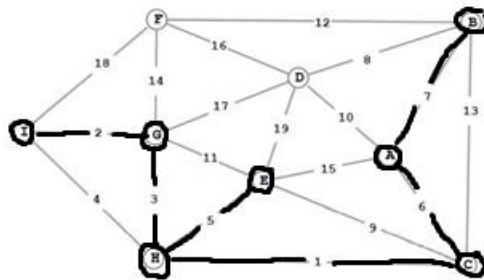
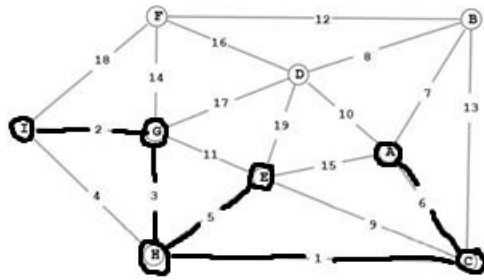
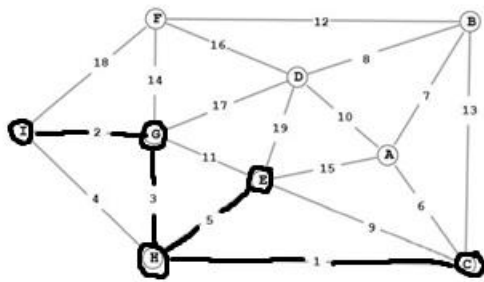






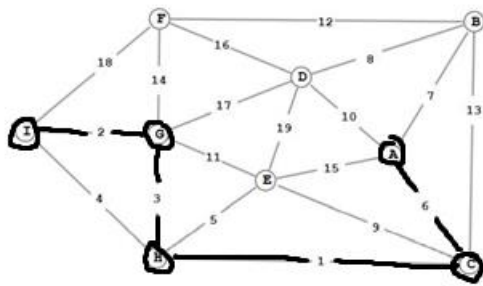
B.Prim(kiri->kanan)





Dijkstra's

I->A



F->C

