1. apa yang dimaksud binary tree?

Binary Tree adalah struktur data yang mirip dengan Linked List. Bila Linked List dianalogikan sebagai rantai yang linier maka Binary Tree dianalogikan sebagai pohon. Binary Tree dikelompokkan menjadi tree yang tidak berurut (unordered Binary Tree) dan tree yang terurut (ordered Binary Tree).Binary tree adalah struktur data yang terdiri dari simpul-simpul yang terhubung secara hierarkis dalam bentuk pohon, di mana setiap simpul memiliki maksimal dua anak, yaitu anak kiri dan anak kanan. Setiap anak kiri memiliki nilai yang lebih kecil dari nilai simpul induknya, sedangkan setiap anak kanan memiliki nilai yang lebih besar.

1. apa yang harus diperhatikan saat membuat program tree?

Saat membuat program tree, beberapa hal yang perlu diperhatikan antara lain:

1. Node atau simpul: Program tree harus memiliki struktur data yang mewakili setiap simpul dalam pohon. Setiap simpul biasanya memiliki data (nilai) dan dua referensi ke anak kiri dan anak kanan (jika menggunakan binary tree).
2. Penyisipan data(insertion): Agar data benar-benar tersusun dalam struktur data Binary Search Tree, dua aturan yang harus dipenuhi pada saat data diatur dalam Binary Search Tree adalah sebagai berikut : Saat menyisipkan data ke dalam pohon, perlu diperhatikan aturan penempatan data agar tetap mempertahankan sifat-sifat binary tree. Misalnya, data yang lebih kecil dari simpul tertentu akan ditempatkan di anak kiri, sedangkan data yang lebih besar akan ditempatkan di anak kanan.
3. Pencarian data(search): Program tree harus mampu melakukan pencarian data berdasarkan nilai tertentu. Pencarian biasanya dilakukan secara rekursif dengan membandingkan nilai pencarian dengan nilai simpul saat ini dan melanjutkan pencarian ke anak kiri atau anak kanan sesuai dengan hasil perbandingan.
4. Traversing(traversal): Traversing atau penelusuran tree digunakan untuk mengunjungi setiap simpul dalam tree. Ada beberapa metode penelusuran yang umum digunakan, seperti inorder, preorder, dan postorder.
5. Menghapus data(deletion): Saat menghapus data dari pohon, perlu diperhatikan aturan penghapusan agar struktur binary tree tetap terjaga. Penghapusan biasanya melibatkan penggantian simpul dengan anak kiri atau anak kanan terdekat, atau penggantian dengan nilai minimum atau maksimum dari anak kiri atau anak kanan.
6. Penanganan kasus khusus: Dalam pembuatan program tree, perlu diperhatikan penanganan kasus khusus seperti penanganan pohon kosong (empty tree), penanganan duplikat data (jika diperbolehkan atau tidak), dan penanganan rotasi atau restrukturisasi tree jika diperlukan.

Selain itu, pemilihan algoritma yang tepat untuk operasi pada tree, seperti pencarian atau penghapusan, juga penting untuk memastikan efisiensi dan konsistensi program tree yang dibuat.