

Nama : Muhammad Galuh Gumelar

Kelas : TPL B1

NIM : J0403221017

1. Jelaskan Pengertian mengenai HeapTree?

HeapTree (juga dikenal sebagai binary heap) adalah struktur data tree (pohon) yang terdiri dari simpul-simpul yang memiliki sifat heap, yaitu setiap simpul memiliki nilai yang lebih kecil atau lebih besar dari nilai simpul anaknya. Secara khusus, binary heap adalah jenis heap yang menerapkan struktur data tree biner.

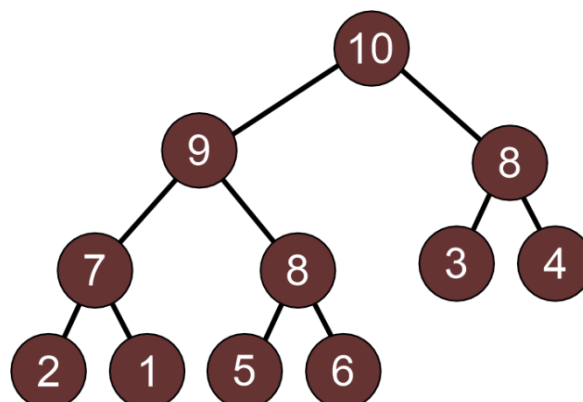
2. Diberikan angka Node berikut ini :

[ 2, 4, 8, 5, 7, 3, 10, 8, 1, 9, 6 ]

a. Bentuklah (konstruksi dan insertion) sebuah heap tree dari node tersebut dengan prioritas pada nilai yang lebih besar! Jelaskan tahapannya

Tahapan :

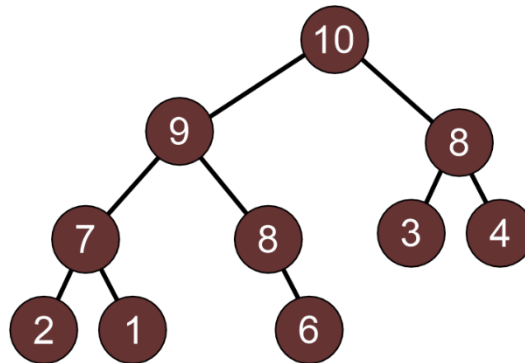
1. Node 2 dijadikan sebagai root utama.
2. Node 4 dimasukkan di sebelah kiri root, karena nilainya lebih besar daripada root. Karena itu, posisi node 4 dan 2 ditukar, sehingga node 4 menjadi parent.
3. Node 8 dimasukkan di sebelah kanan root.
4. Node 5 dimasukkan di sebelah kiri node 4, karena nilainya lebih besar daripada node 2. Karena itu, posisi node 5 dan 2 ditukar, sehingga node 5 menjadi parent.
5. Node 7 dimasukkan di sebelah kanan node 5, karena nilainya lebih besar daripada node 4. Karena itu, posisi node 7 dan 5 ditukar, sehingga node 7 menjadi parent.
6. Node 3 dimasukkan di sebelah kiri node 8, karena nilainya lebih kecil daripada node 8.
7. Node 10 dimasukkan di sebelah kanan node 8, karena nilainya lebih besar daripada node 2. Karena itu, posisi node 10 dan 2 ditukar, sehingga node 10 menjadi parent dan menjadi root utama. Node 10 juga bertukar posisi dengan node 4.
8. Node 8 dimasukkan di sebelah kiri node 2, karena nilainya lebih besar daripada node 7. Karena itu, posisi node 8 dan 2 ditukar, sehingga node 8 menjadi parent dari node 7.
9. Node 1 dimasukkan di sebelah kiri node 8, karena nilainya lebih kecil daripada node 8.
10. Node 9 dimasukkan di sebelah kiri node 5, karena nilainya lebih besar daripada node 4. Karena itu, posisi node 9 dan 4 ditukar, sehingga node 9 menjadi parent dari node 8.
11. Node 6 dimasukkan di sebelah kanan node 8, karena nilainya lebih kecil daripada node 8.



- b. Jelaskan Proses penghapusan terhadap Heaptree dari node tersebut (berikan contoh) dan Jelaskan tahapanya.

Tahapan :

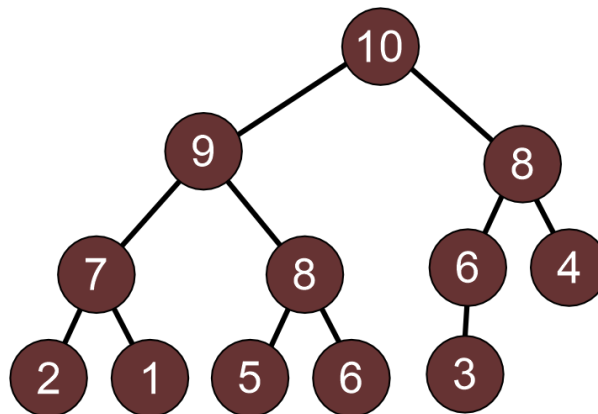
1. Hapus node 5
2. Dikarenakan node 5 tidak memiliki child, maka tidak masalah untuk langsung dihapus



- c. Jelaskan proses penyisipan pada Heaptree dari node tersebut (berikan contoh) dan Jelaskan tahapanya

Tahapan :

- 1) Menyisipkan node 6
- 2) Node 6 ditempatkan pada child kiri dari parent 3
- 3) Node 6 dan 3 bertukar posisi karena lebih besar nilainya, sehingga Node 6 menjadi parent dari Node 3



3. Apa yang dimaksud dengan Binary Search Tree?

Binary Search Tree (BST) adalah struktur data pohon yang terdiri dari simpul-simpul (node) yang diurutkan sedemikian rupa sehingga setiap simpul memiliki nilai yang lebih besar dari nilai simpul di bawahnya pada cabang kiri dan nilai yang lebih kecil dari simpul di bawahnya pada cabang kanan. Dalam BST, setiap node hanya memiliki maksimal dua cabang, yaitu cabang kiri dan cabang kanan, dan setiap cabang menampung node-node yang memiliki nilai yang lebih kecil atau lebih besar dari nilai node induknya.

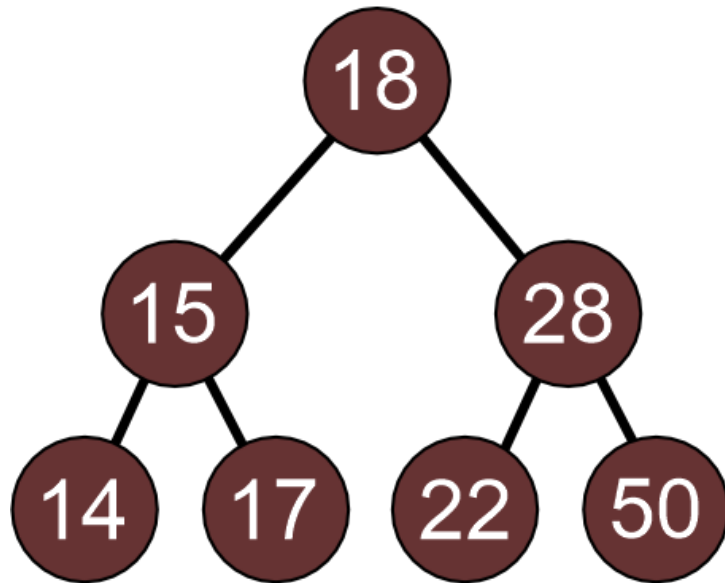
4. Diberikan angka node berikut ini :

[ 18, 28, 15, 22, 50, 14, 17 ]

a. Buatlah sebuah BST dari node tersebut? Jelaskan tahapannya

Tahapan:

1. Node pertama dengan nilai 18 dijadikan sebagai root.
2. Node kedua dengan nilai 28 dibandingkan dengan root, karena nilai 28 lebih besar maka diletakkan di sisi kanan root sebagai child.
3. Node ketiga dengan nilai 15 dibandingkan dengan root, karena nilai 15 lebih kecil maka diletakkan di sisi kiri root sebagai child.
4. Node keempat dengan nilai 22 dibandingkan dengan root dan parent yang bernilai 28, karena nilai 22 lebih kecil dari 28 maka diletakkan di sisi kiri node 28 sebagai child.
5. Node kelima dengan nilai 50 dibandingkan dengan root dan parent yang bernilai 28, karena nilai 50 lebih besar dari 28 maka diletakkan di sisi kanan node 28 sebagai child.
6. Node keenam dengan nilai 14 dibandingkan dengan root dan parent yang bernilai 15, karena nilai 14 lebih kecil dari 15 maka diletakkan di sisi kiri node 15 sebagai child.
7. Node ketujuh dengan nilai 17 dibandingkan dengan root dan parent yang bernilai 15, karena nilai 17 lebih besar dari 15 maka diletakkan di sisi kanan node 15 sebagai child.



b. Lakukan penghapusan node pada nilai 8 dan Jelaskan tahapanya

Tahapan :

1. Periksa apakah node yang akan dihapus memiliki dua child. Jika ya, pilih node terbesar pada subtree kiri atau node terkecil pada subtree kanan untuk menggantikan posisinya.
2. Karena node 18 memiliki dua child (yaitu node 15 dan node 28), maka kita perlu mencari node terbesar pada subtree kiri atau node terkecil pada subtree kanan. Karena subtree kiri hanya memiliki node dengan nilai 15, maka kita perlu mencari node terbesar pada subtree kanan, yaitu node 22.
3. Setelah kita menemukan node 22, kita pindahkan node tersebut untuk menggantikan posisi root, sehingga node 22 menjadi root baru.

