

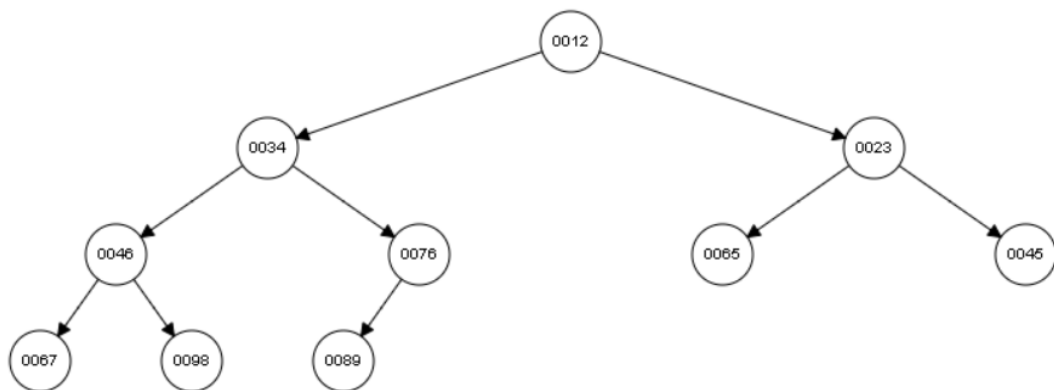
Problem Set 02 Solution

Nama : Ikhwanul Abiyu Dhiyya'ul Haq

NRP : 5024211048

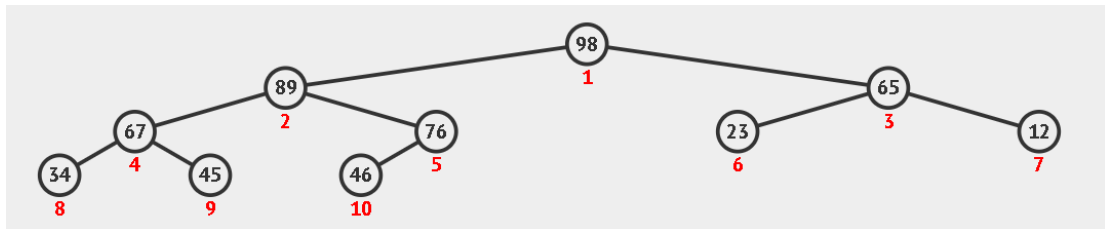
Problem Set Tree

1. Tree kiri merupakan min-heap, sedangkan tree kanan bukan min-heap. Syarat dari min-heap yaitu root node harus kurang atau sama dengan key pada semua child-nya. Pada tree kanan, angka 20 yang bertindak sebagai internal node, memiliki child yang lebih kecil darinya, yaitu 18. Maka tree kanan tidak termasuk min-heap.
2. Tree kanan maupun kiri merupakan max-heap. Syarat dari max-heap yaitu root node harus lebih besar atau sama dengan key pada semua child-nya.
3. Pada array 45, 67, 23, 46, 89, 65, 12, 34, 98, 76. Jika dibuat min-heap, maka 45 akan diinsert terlebih dahulu dan menjadi Root dari tree. Kemudian 67 diinsert, ke kiri terlebih dahulu, dicek apakah yang diinsert lebih kecil dari parentnya? Jika iya, maka swap, jika tidak maka biarkan. Dalam hal ini 67 lebih besar dari 45, sehingga tidak perlu diswap. Kemudian 23 di-insert, ke kanan. Dicek kembali apakah 23 lebih kecil dari 45 selaku parent? Jika iya maka diswap, jika tidak maka biarkan. Dalam case ini, 23 lebih kecil dari 45, sehingga 23 dan 45 bertukar posisi, sehingga 23 menjadi parent dari tree ini. Begitu seterusnya hingga data terakhir. Hasil akhir dari min-heap tersebut seperti pada gambar di bawah ini.



4. Pada array 45, 67, 23, 46, 89, 65, 12, 34, 98, 76. Jika dibuat max-heap, maka 45 akan diinsert terlebih dahulu dan menjadi Root dari tree. Kemudian 67 diinsert, ke kiri terlebih dahulu, dicek apakah yang diinsert lebih besar dari parentnya? Jika iya, maka swap, jika tidak maka biarkan. Dalam hal ini 67 lebih besar dari 45, sehingga 67 naik menjadi parent dari tree. Kemudian 23 di-insert, ke kanan. Dicek kembali apakah 23 lebih besar dari 67 selaku parent? Jika iya maka diswap, jika tidak maka biarkan. Dalam

case ini, 23 lebih kecil dari 67, sehingga tidak terjadi apa-apa. Begitu seterusnya hingga data terakhir. Hasil akhir dari max-heap tersebut seperti pada gambar di bawah ini.



5. Langkah-langkah dalam mendelete semua node hingga menjadi empty max-heap yaitu:
Di sini saya mendelete satu persatu dari child paling bawah sesuai urutan datanya. Yaitu dimulai dari 13. 13 dijadikan parent + 1, sehingga dia akan naik (diswap) hingga ke parent. Kemudian setelah menjadi parent, parent + 1 ini di-delete. Lalu child terakhir setelah 13 di-delete tadi (yaitu 12). Naik menjadi parent. Lalu sesuai dengan prinsip max-heap tadi. 12 akan dicek, apakah lebih kecil dari childnya, diprioritaskan kanan terlebih dahulu, jika kanan lebih tidak lebih kecil, maka dicek kirinya. Hingga angka 12 ke tempat seharusnya. Begitu seterusnya hingga max-heap kosong.
6. Untuk nomer 6 akan saya representasikan dalam tabel

Initial Condition

Array

7	11	9	23	41	27	12	29
---	----	---	----	----	----	----	----

Tree

--	--	--	--	--	--	--	--

Insert 7

Array

7	11	9	23	41	27	12	29
---	----	---	----	----	----	----	----

Tree

7							
---	--	--	--	--	--	--	--

Insert 11

Array

7	11	9	23	41	27	12	29
---	----	---	----	----	----	----	----

Tree

7	11						
---	----	--	--	--	--	--	--

Insert 9

Array

7	11	9	23	41	27	12	29
---	----	---	----	----	----	----	----

Tree

7	11	9					
---	----	---	--	--	--	--	--

Insert 23

Array

7	11	9	23	41	27	12	29
---	----	---	----	----	----	----	----

Tree

7	11	9	23				
---	----	---	----	--	--	--	--

Insert 41

Array

7	11	9	23	41	27	12	29
---	----	---	----	----	----	----	----

Tree

7	11	9	23	41			
---	----	---	----	----	--	--	--

Insert 27

Array

7	11	9	23	41	27	12	29
---	----	---	----	----	----	----	----

Tree

7	11	9	23	41	27		
---	----	---	----	----	----	--	--

Insert 12

Array

7	11	9	23	41	27	12	29
---	----	---	----	----	----	----	----

Tree

7	11	9	23	41	27	12	
---	----	---	----	----	----	----	--

Insert 29

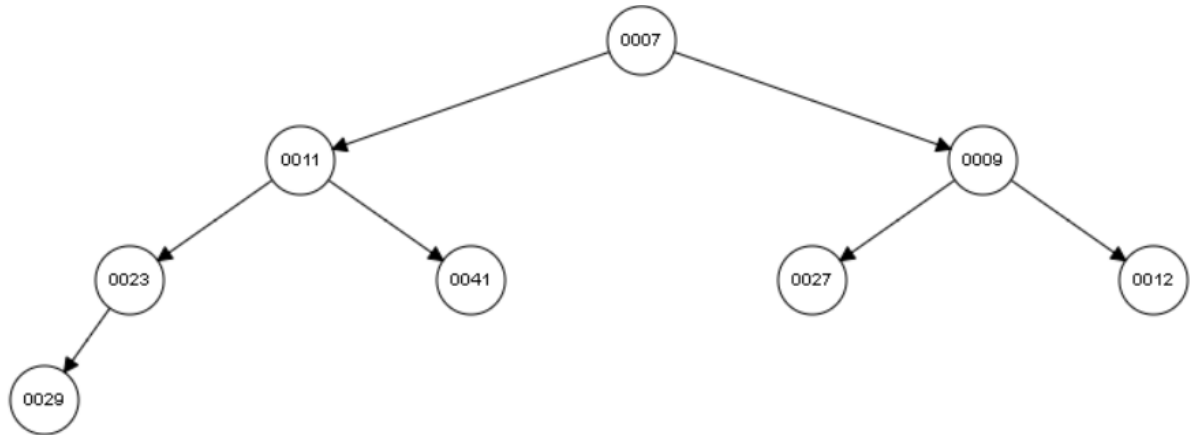
Array

7	11	9	23	41	27	12	29
---	----	---	----	----	----	----	----

Tree

7	11	9	23	41	27	12	29
---	----	---	----	----	----	----	----

Hasil akhir



Next Step : Insert 6 and 55

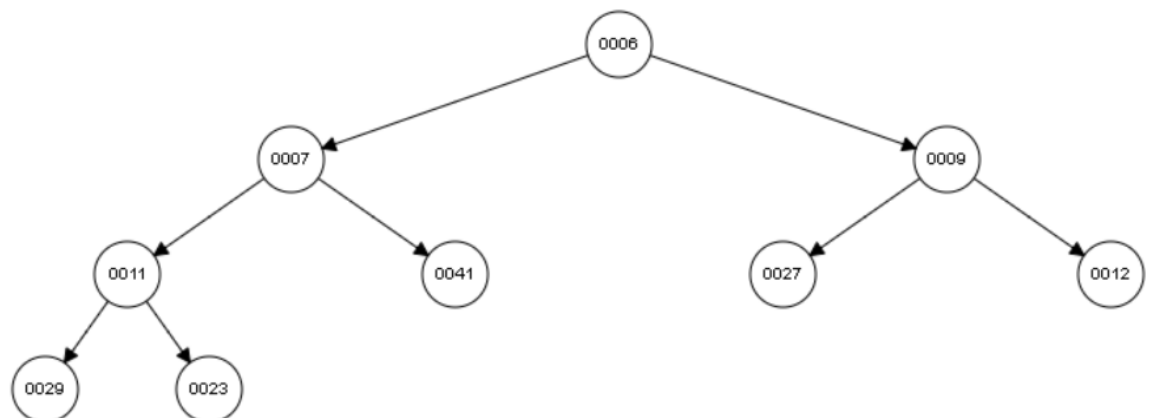
Insert 6

Array

7	11	9	23	41	27	12	29
---	----	---	----	----	----	----	----

Tree

6	7	9	11	41	27	12	29	23
---	---	---	----	----	----	----	----	----



Insert 55

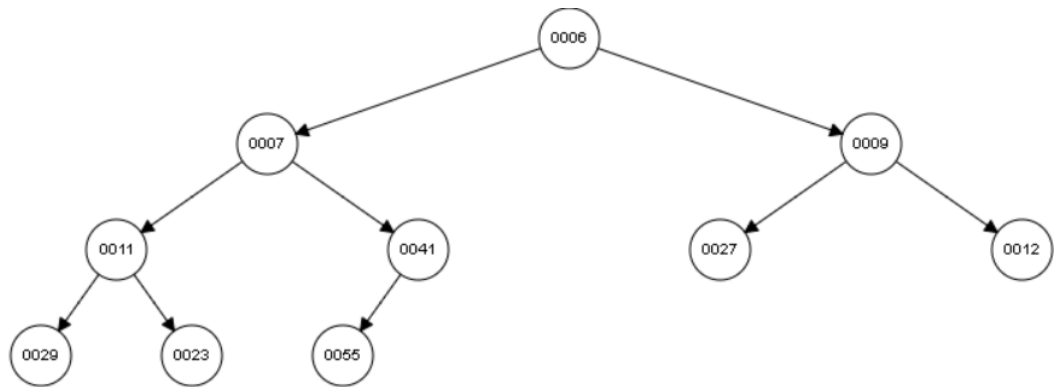
Array

7	11	9	23	41	27	12	29
---	----	---	----	----	----	----	----

Tree

6	7	9	11	41	27	12	29	23	55
---	---	---	----	----	----	----	----	----	----

Hasil akhir



Next step : Delete until empty min-heap

Delete 55

Step 1

Tree

6	7	9	11	41	27	12	29	23	55
---	---	---	----	----	----	----	----	----	----

Step 2

Tree

6	7	9	11	41	27	12	29	23	5
---	---	---	----	----	----	----	----	----	---

Step 3

Tree

6	7	9	11	5	27	12	29	23	41
---	---	---	----	---	----	----	----	----	----

Step 4

Tree

6	5	9	11	7	27	12	29	23	41
---	---	---	----	---	----	----	----	----	----

Step 5

Tree

5	6	9	11	7	27	12	29	23	41
---	---	---	----	---	----	----	----	----	----

Step 6

Tree (with no parent after delete 5)

6	9	11	7	27	12	29	23	41
---	---	----	---	----	----	----	----	----

Step 7

Tree

41	6	9	11	7	27	12	29	23
----	---	---	----	---	----	----	----	----

Step 8

Tree

6	41	9	11	7	27	12	29	23
---	----	---	----	---	----	----	----	----

Step 9

Tree

6	11	9	41	7	27	12	29	23
---	----	---	----	---	----	----	----	----

Step 10

Tree

6	11	9	29	7	27	12	41	23
---	----	---	----	---	----	----	----	----

Dan seterusnya.