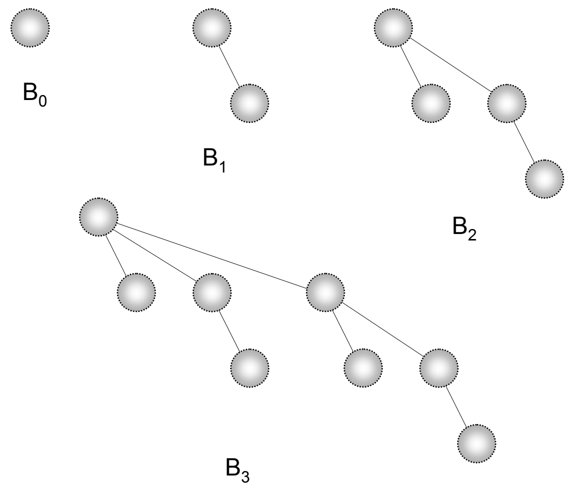
Priority Queues - 2

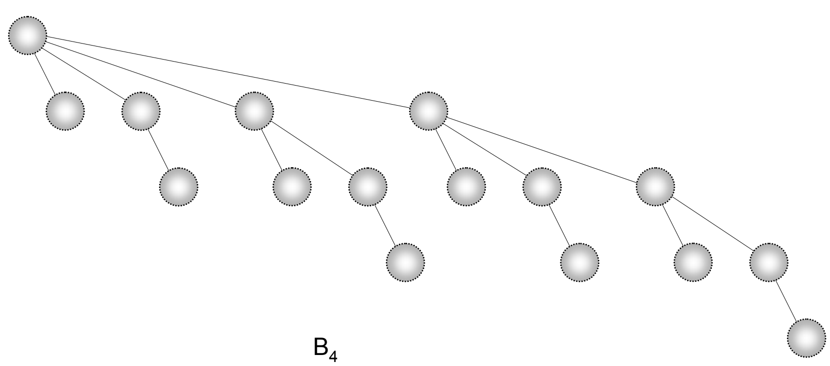
CS 202 – Fundamental Structures of Computer Science II
Bilkent University
Computer Engineering Department

Binomial Queues

- Binamial Queue
  - A collection of heap-ordered trees (a forest).
  - A tree in the forest is a *binomial tree*.
  - There is *at most one* binomial tree of *every height* in the forest.
- A binomial tree of height 0 has one node.
- A binomial tree of height k has 2^k nodes.
- A binomial tree of height k, B_k, is formed by attaching a binomial tree of height k-1, B_{k-1}, to another binomial tree of height k-1, B_{k-1}.

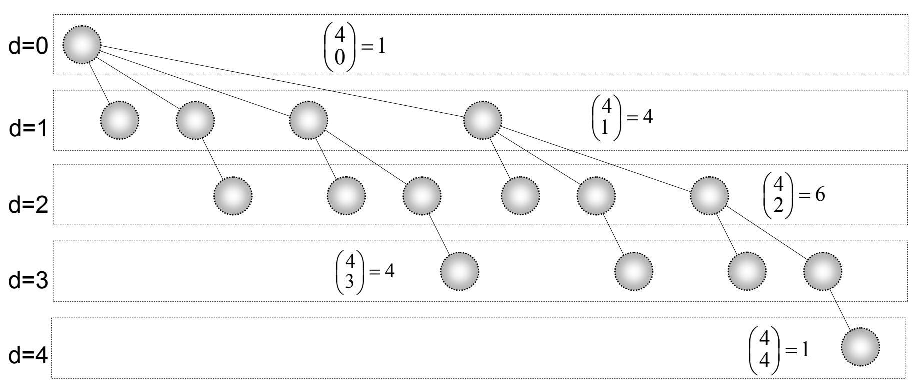
Binomial Trees





- A binomial tree B_k consists of root with children B_0, B_1, …, B_{k-1}
- Binomial trees of height **k** have exactly 2^k nodes.
- The number of nodes at depth *d* is binomial coefficient $\binom{k}{d}$

k = 4

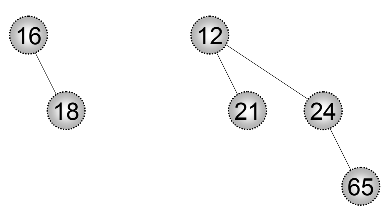


B_4

Heaps and binomial trees

- Assume all binary trees in a forest are in heap order.
- Assume we have at most one binomial tree of any height.
- Then we can represent a heap of any size uniquely by a binomial tree forest.
- Example: heap of size 13 could be represented by forest: B_3, B_2, B_0,
- We call this heap ordered binomial forest a binomial queue.

Example



H_1

A binomial queue H_1 of size 6 is shown above

Can be represented as: 110

Binomial Queue Operations

- FindMin
- Merge
- Insert
- DeleteMin

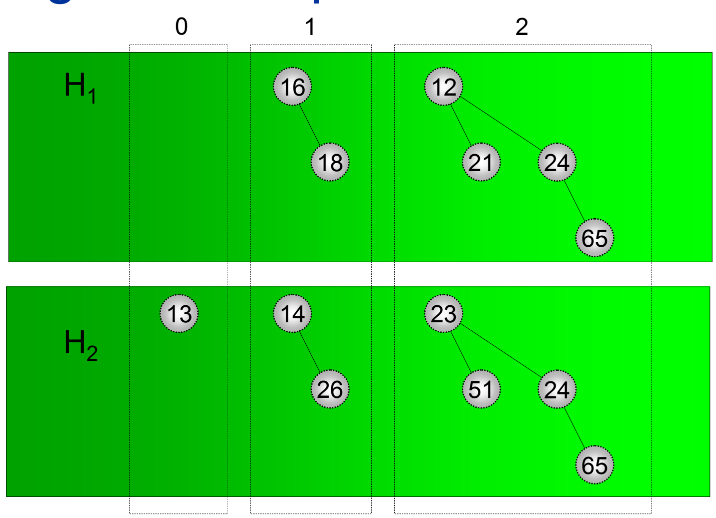
FindMin

- The minimum element in the binomial queue can be found by scanning the roots of all trees in the forest.
- There are at most logN different trees
- The cost of findMin is therefore O(logN) in the worst case.

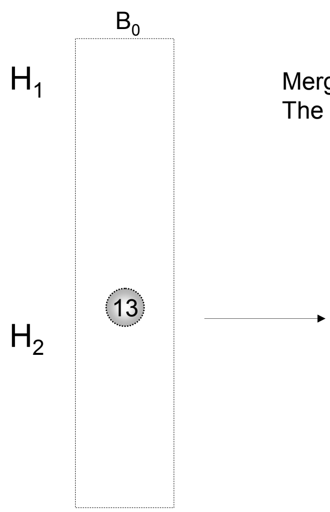
Merge

- Merging two binomial queues is conceptually simple.
- Merge H_1 and H_2
- Just add them in binary.
- Assume H_1 has 6 nodes. H_2 has 7 nodes.
  - H_3 = node (H_1) + nodes(H_3) = 13.
  - H_1: 0110
  - H_2: 0111
  - H_3: 1101 (this is the resulting bin. heap).

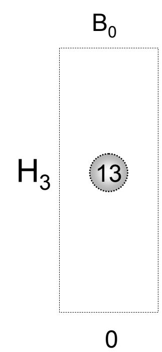
Merge - example



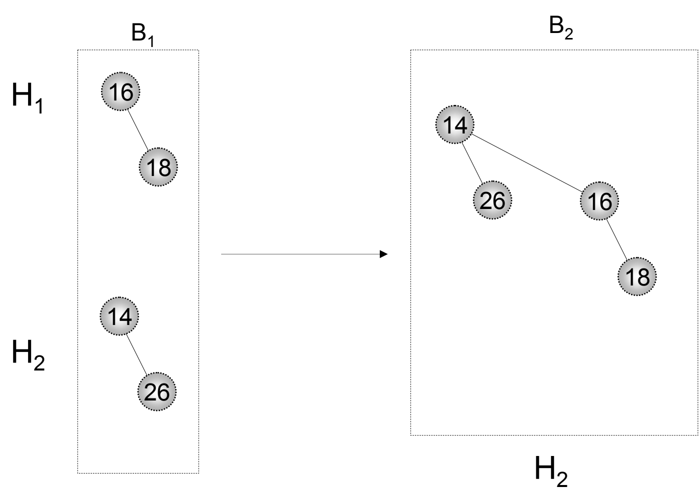
Merge - example - steps

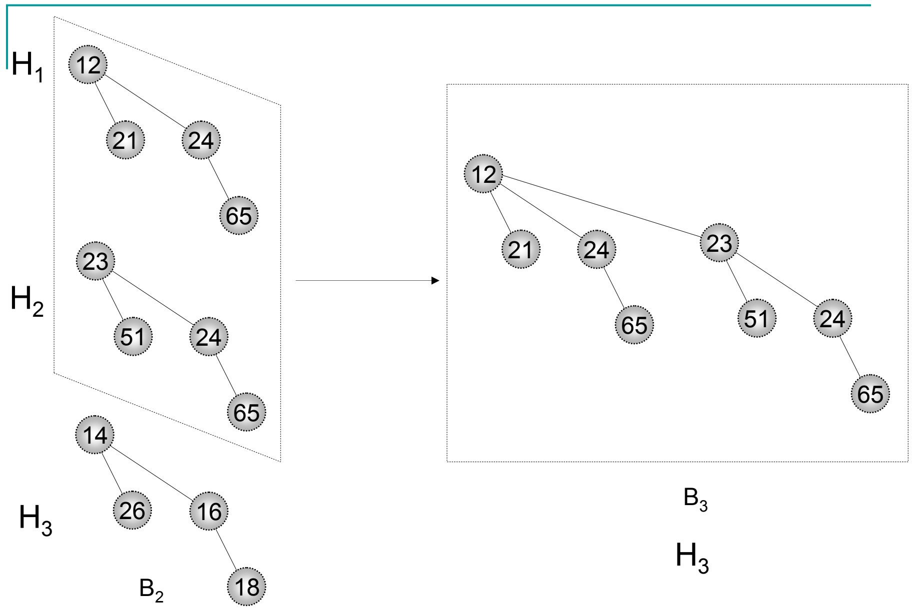


Merge 1 node tree with empty tree.
The result is the one node tree itself.

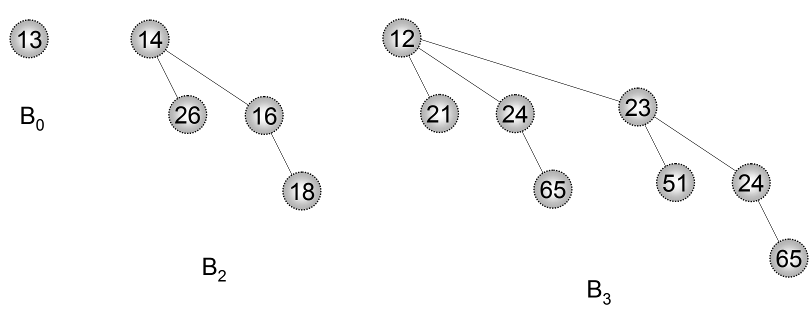


Merge - example - steps





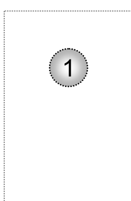
Final Binomial Queue H_3



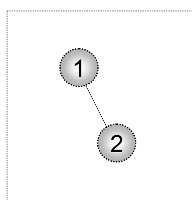
$H_3 = 1101$

Insert

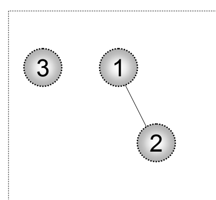
- Insertion is a special case of merge
- Insert an item to a binomial heap H_1
- Make a new heap, H_2, of one node (item to be inserted)
- Merge H_1 and H_2.
- Example
  - Insert items 1, 2, 3, 4, 5, 6, 7, in the given order into an empty binomial heap.



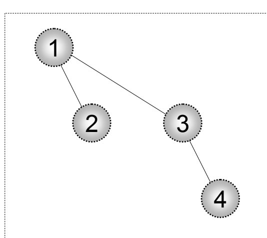
After 1 is inserted



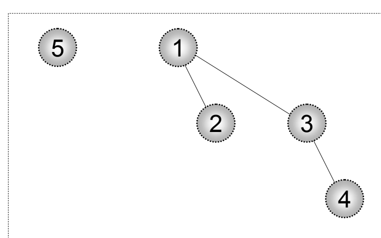
After 2 is inserted



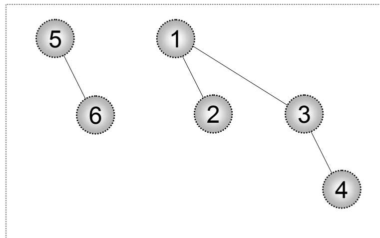
After 3 is inserted



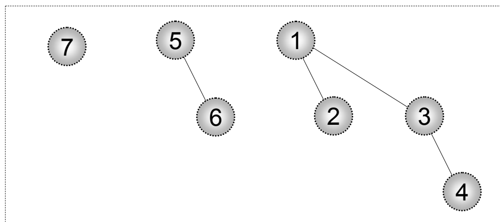
After 4 is inserted



After 5 is inserted



After 6 is inserted

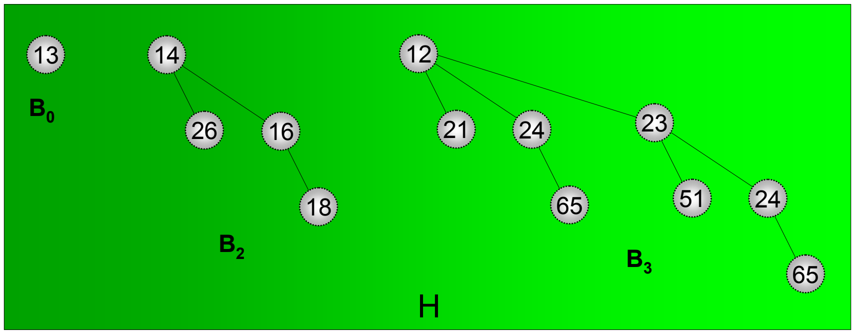


After 7 is inserted

DeleteMin

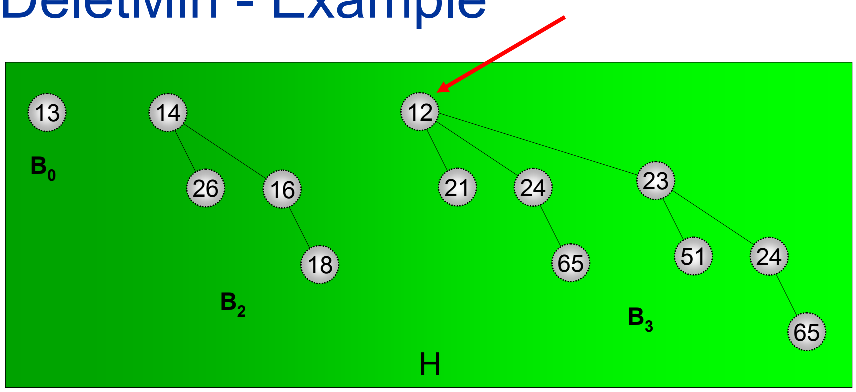
- Sketch of the Algorithm
- Assume we want to delete minimum item from binomial queue H.
  - Find the binomial tree in H that has the minimum root. Let say this is B_k in H.
  - Take tree B_k from H. Let the remaining trees in H make a new binomial queue H_1
  - Remove the root from B_k (root is the minimum that you should return as a result of algorithm).
    - The children of root of B_k make a new heap H_2 that can consists of tree B_0 through B_{k-1}
  - Merge H_1 and H_2.

DeletMin - Example

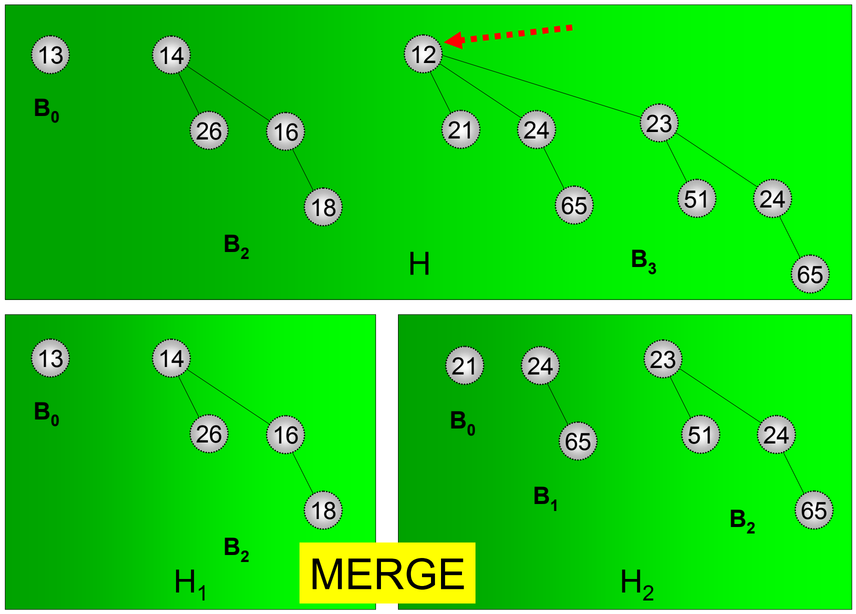


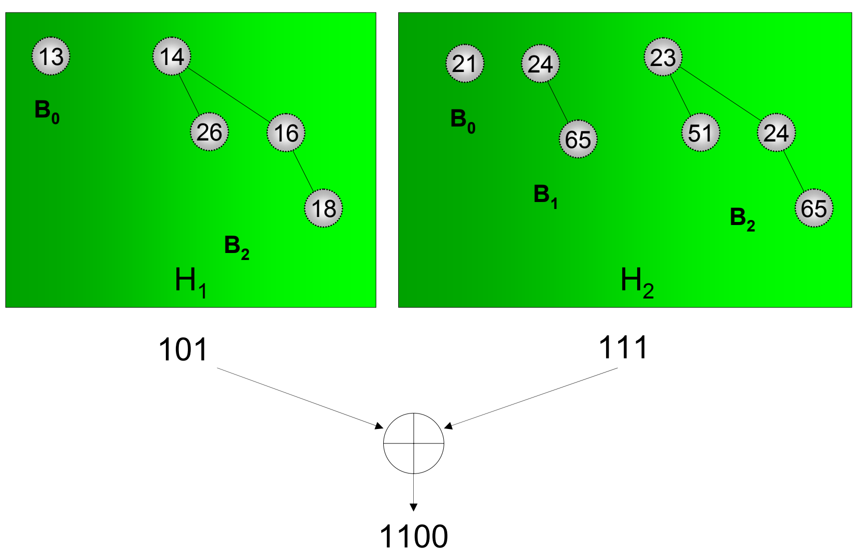
Delete the minimum item from the binomial queue above.

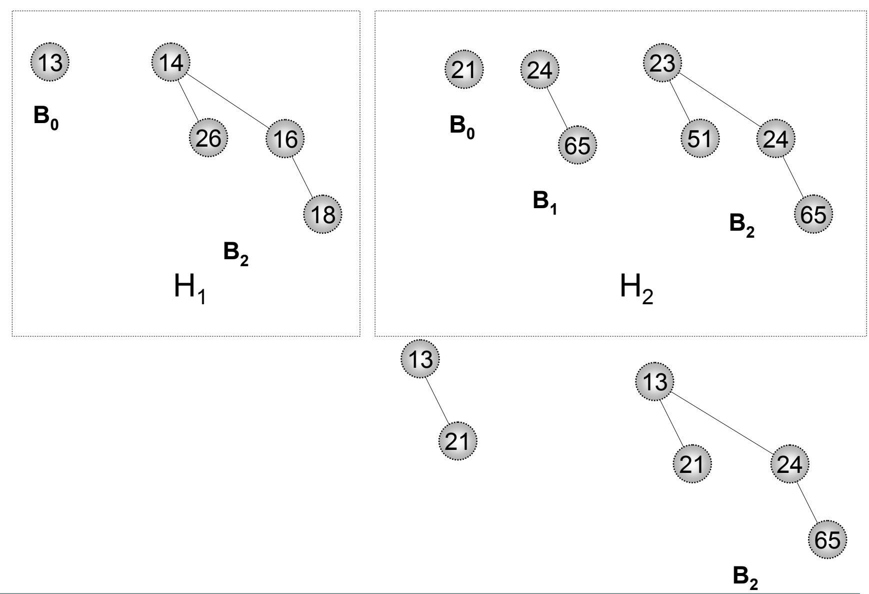
DeletMin - Example

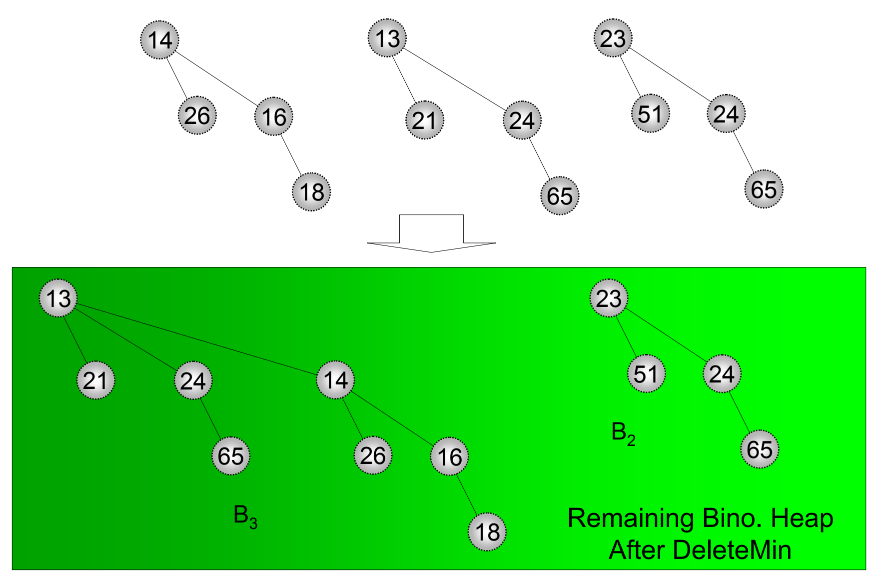


The root that has the minimum item is 12 and belongs to tree B_3


13
14
26
16
18
12
21
24
65
23
51
24
65
B0
B2
H
B3
13
14
26
16
18
B0
B2
H1
MERGE
21
24
65
23
51
24
65
B0
B1
B2
H2


13
14
26
16
18
B0
B2
H1
21
24
65
23
51
24
65
B0
B1
B2
H2
101
111
1100


13
B0
14
26
16
18
B2
H1
21
24
23
B0
65
51
24
B1
B2
65
H2
13
21
13
21
24
65
B2


14
26
16
18
13
21
24
65
23
51
24
65
13
21
24
65
B3
14
26
16
18
23
51
24
B2
65
Remaining Bino. Heap
After DeleteMin