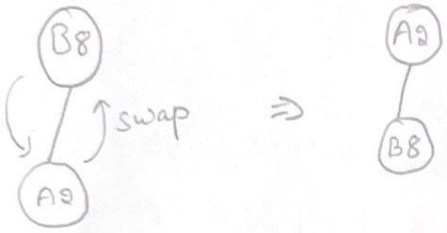
Name Pahuldeep Singh, Student id:3153555

Passenger Comparator

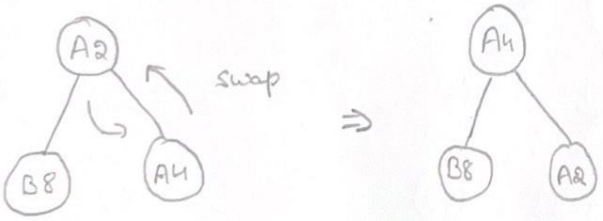
① standBy list: B8 addition

② List A2 addition:



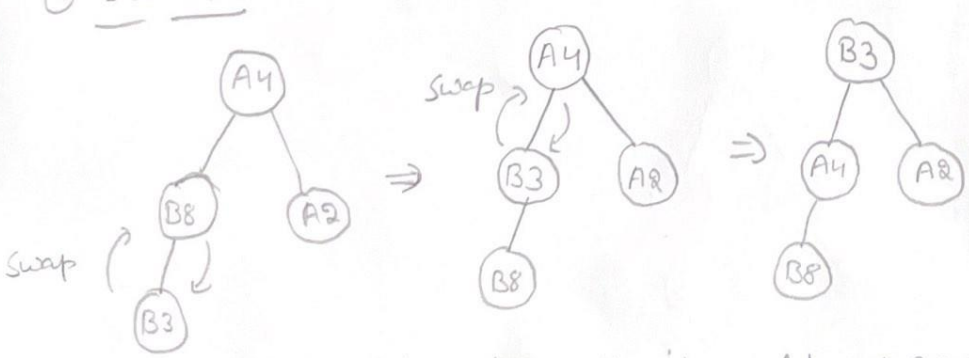
Swap occured due to heap order property, since A2 has the small key than B8 and A2 becomes the root.

③ A4 addition:



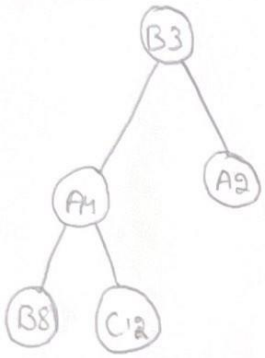
A4 added at right child to maintain complete binary tree property, Swap occured since A4 has the lowest key than A2 and it would result in root due to heap order property.

④ B3 addition:



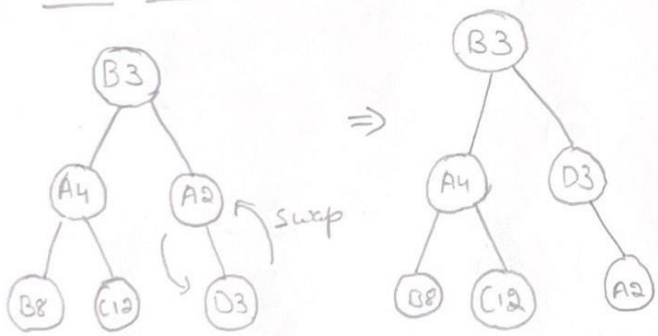
Since B3 has the lowest priority, it would get swapped with B8 and then A4 due to heap order property, it becomes the root as well

(5) C12 Addition:



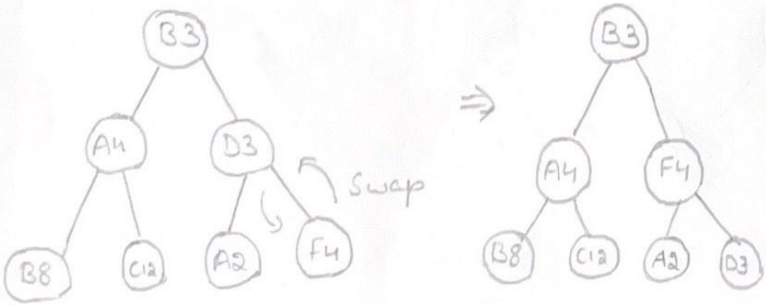
Since C12 has highest key, it will add at the right child of A4 to maintain heap order and complete binary tree property. So swapping occurs.

(6) D3 Addition:



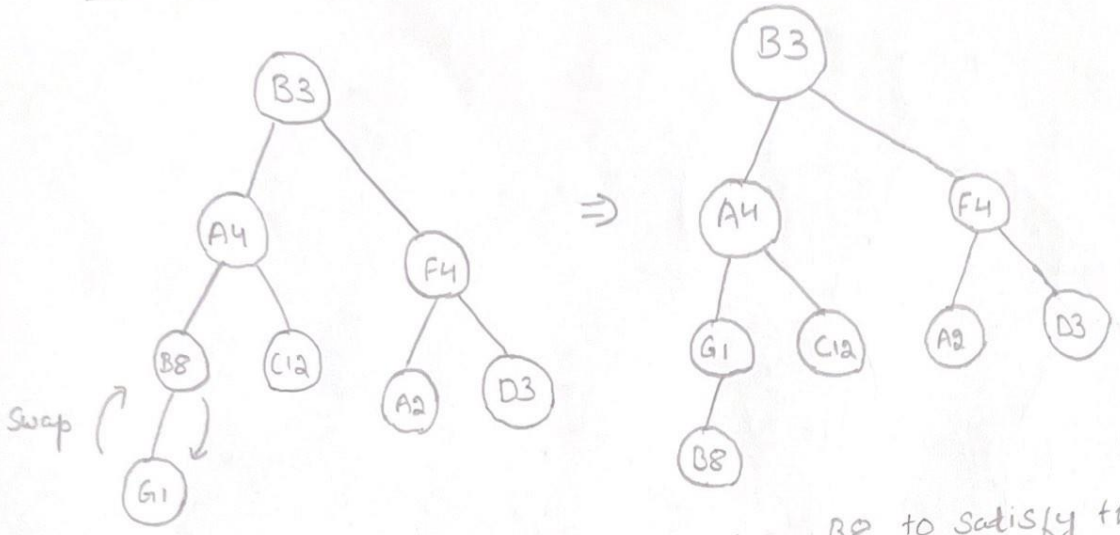
D3 gets swapped with A2 due to lower key than A2. (Heap order property).

(7) F4 Addition:



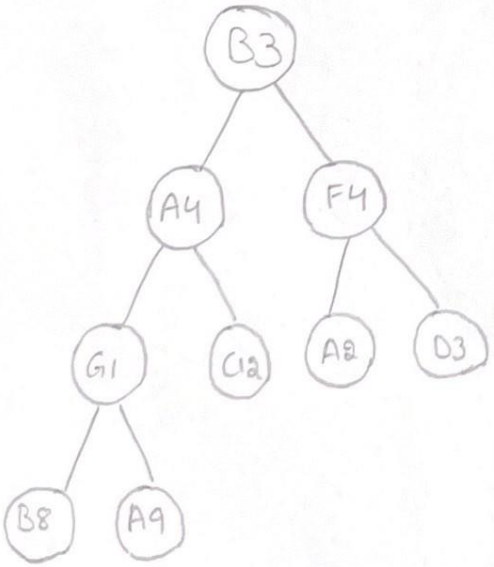
Swapping occured with D3 by F4, because F4 has the small key than D3. (Heap order property).

(8) G1 Addition:



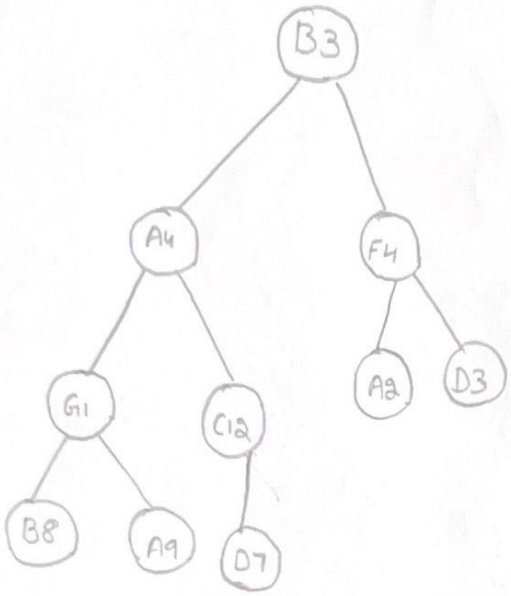
G1 will get added at the left child of B8 to satisfy the complete Binary tree property, then it gets swapped with B8 due to heap order property of heap.

(9) A9 addition:



A9 will get added as the right child, it will not get swapped to maintain heap order property.

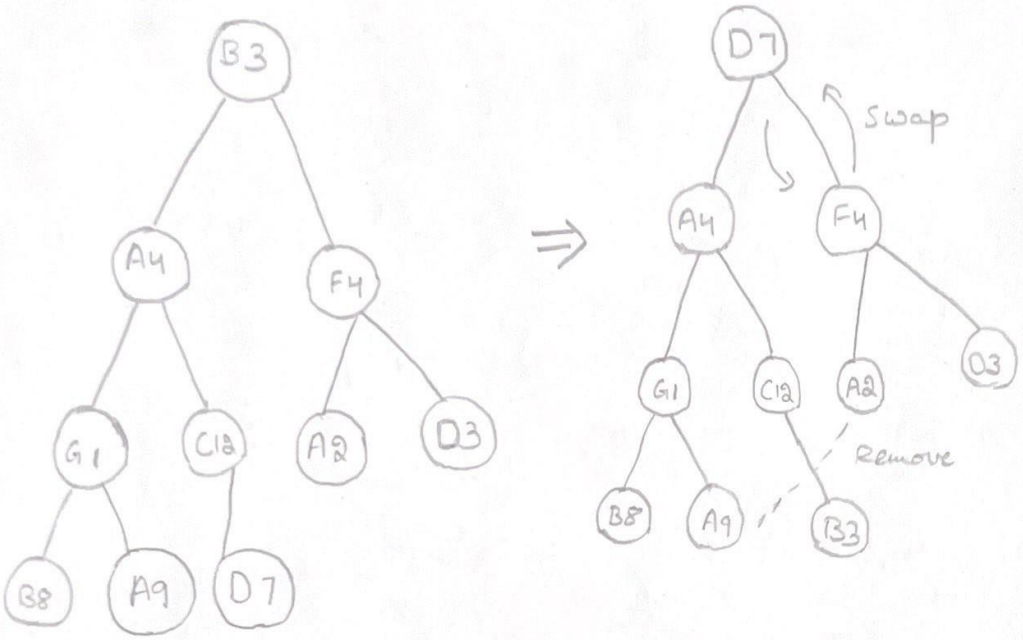
(10) D7 Addition:

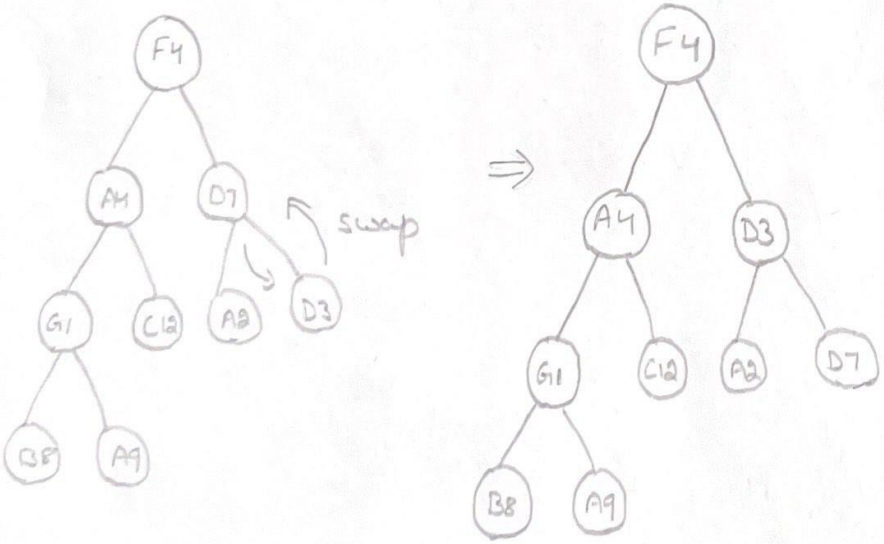


D7 will get added to C12 as its left child will not engage in any swapping due to heap order property as D7 has high key than C12.

PassengerComparator

① Removing B3

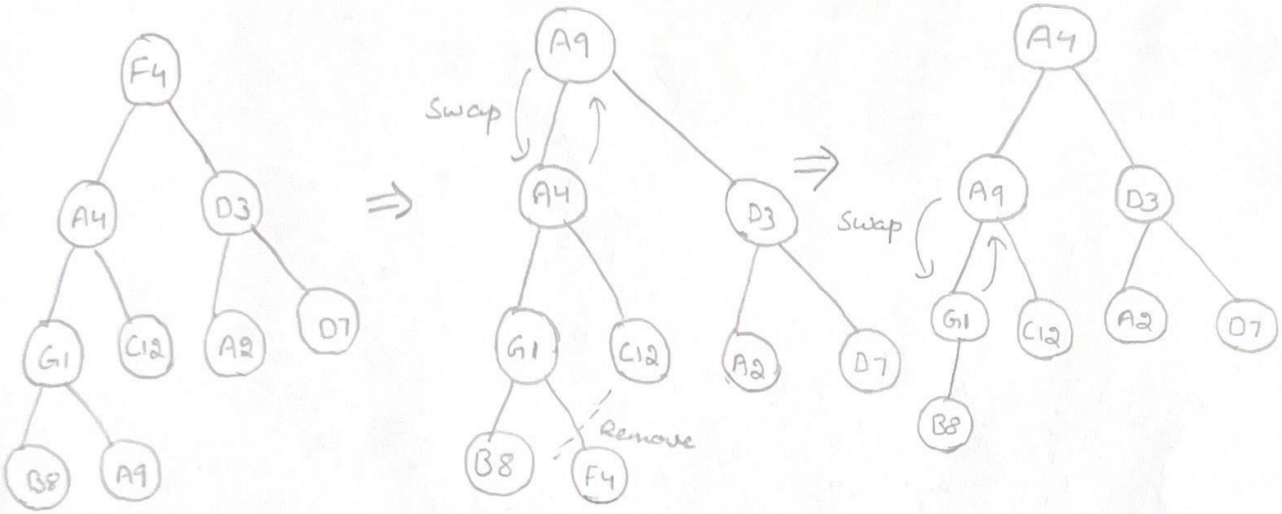


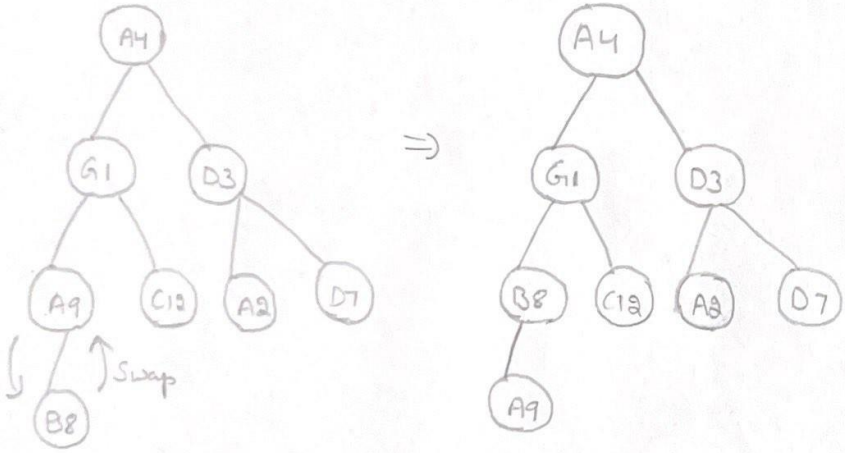


Since B3 was at root and it had minimal key and was to be removed, after upheaping process D7 was at the root and B3 gets removed, to maintain heap order property D7 gets swapped with F4 and then it compares its key with its left and right child and swapes with lower key child D3 due to heap order property

PassengerComparator

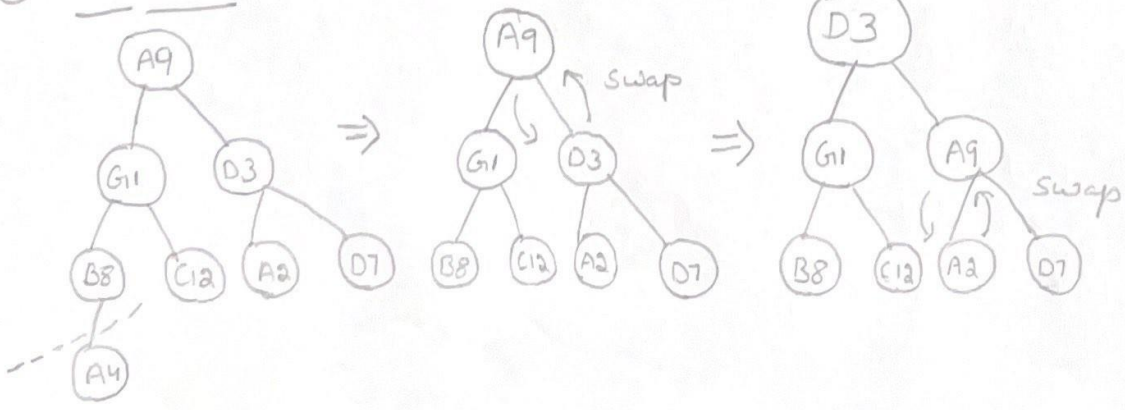
(2). F4 Removed:

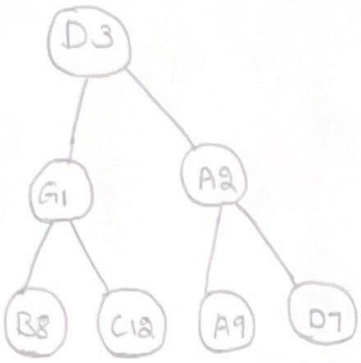




Since A4 had lowest key than A9, it gets swapped with A9, A9 then compares it key with its both child, it then swapped with lowest key child G1. A9 then gets swapped with B8 to maintain heap order property.

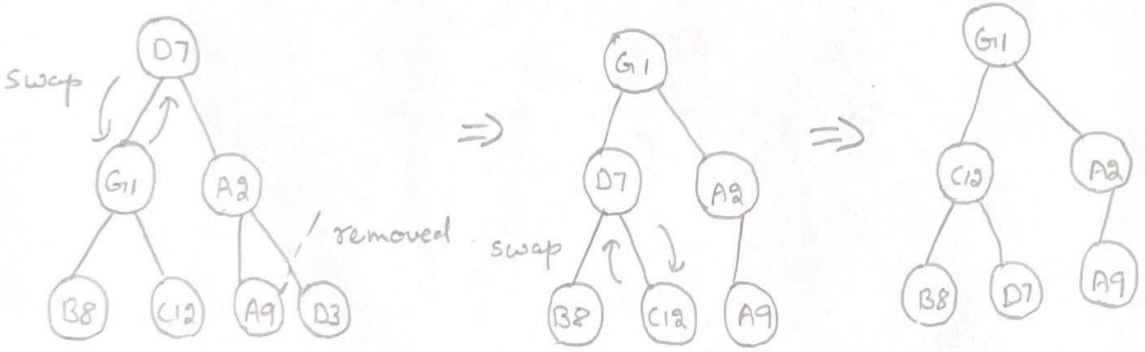
(3) A4 Removed:-





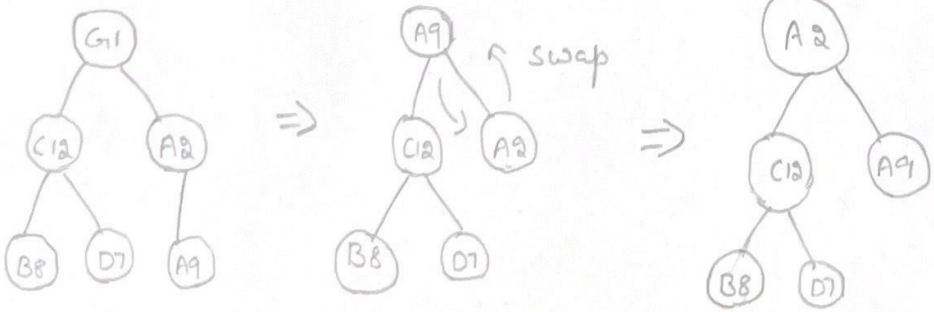
A4 gets removed, D3 became the root after A9 compares its key with D3. Moreover, A9 then compares its key with its child A2 and gets swapped to maintain heap order property.

(4) D3 removed :-



Since, G1 had the lowest key it gets swapped with D7 and becomes root. D7 had high key than C12, so it gets swapped to maintain heap order property.

(5) G1 Removed :-



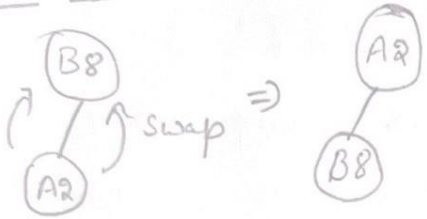
A9 gets swapped with A2 to maintain heap order property.

① Default comparator :-

Default Comparator

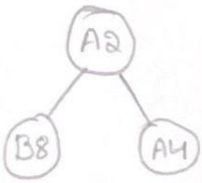
B8 :

② A2 Addition :-



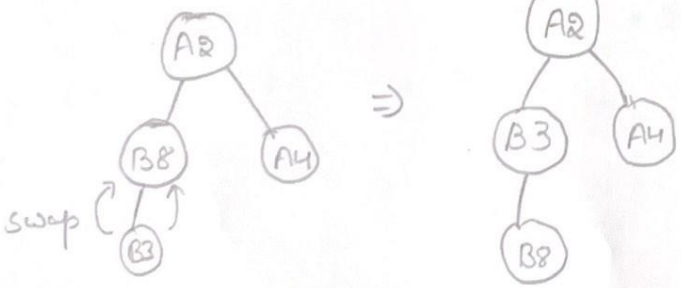
A2 gets swapped with B8 due to heap order property because A2 had low key.

③ A4 addition :



A4 added as a right child of A2 due to complete binary tree property.

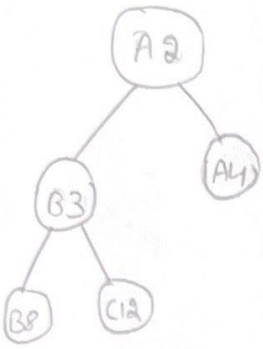
④ B3 addition :



B3 added as left child of B8, it then get swapped with B8 to maintain heap order property.

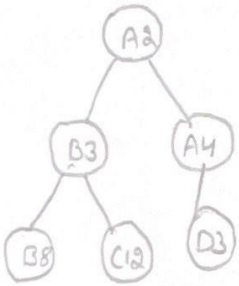
⑤ C12 Addition:

Default comparator



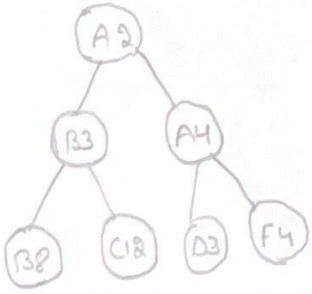
C12 added as a right child to B3. No swapping would occur due to highest key.

⑥ D3 Addition:



D3 added as Left child of A4, No swapping would occurs since it is highest key.

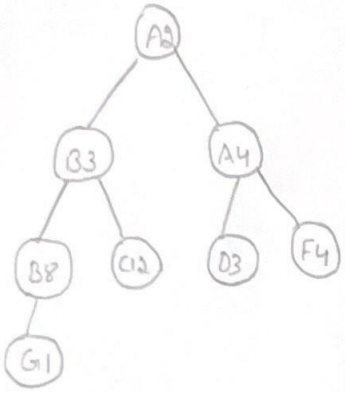
⑦ F4 Addition!



F4 added as Right child of A4, no swapping would occur to main heap order property.

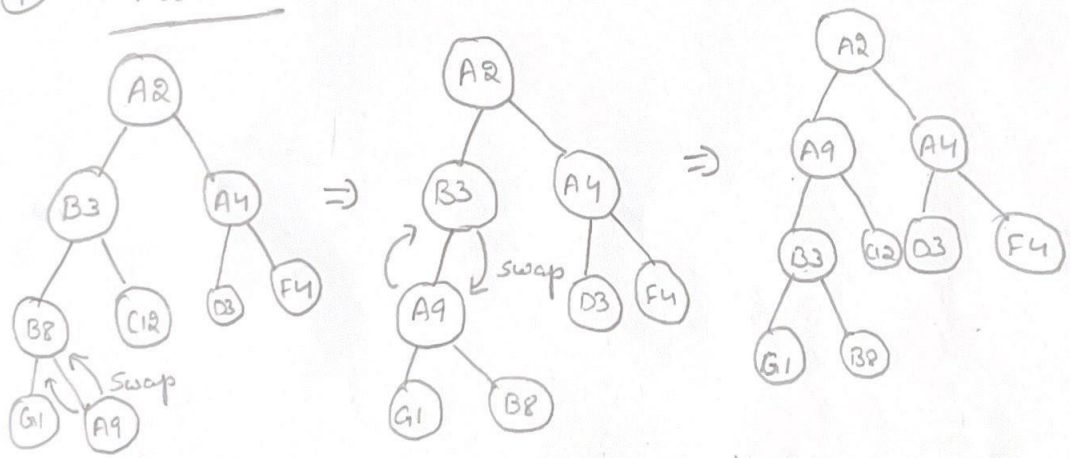
(8) G1 Addition:

Default Comparator



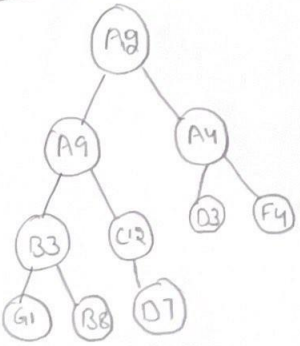
G1 added as left child of B8 to maintain complete binary tree property and heap order property.

(9) A9 addition:



A9 added at the right child of B8, it gets swapped with B8, then with B3 due to heaporder property.

(10) D7 Addition:

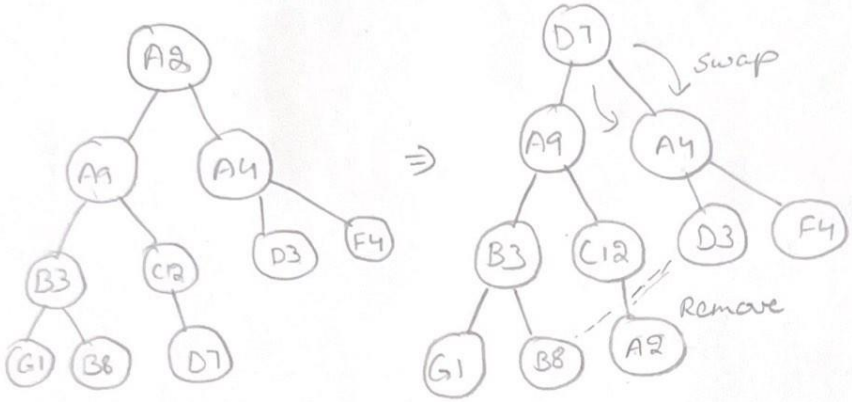


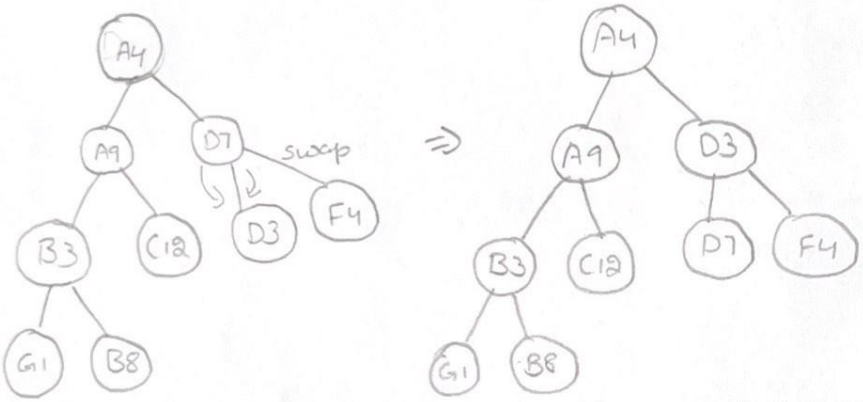
No swapping would occur, since it is high key due to heap order property.

Default Comparator

Removations:

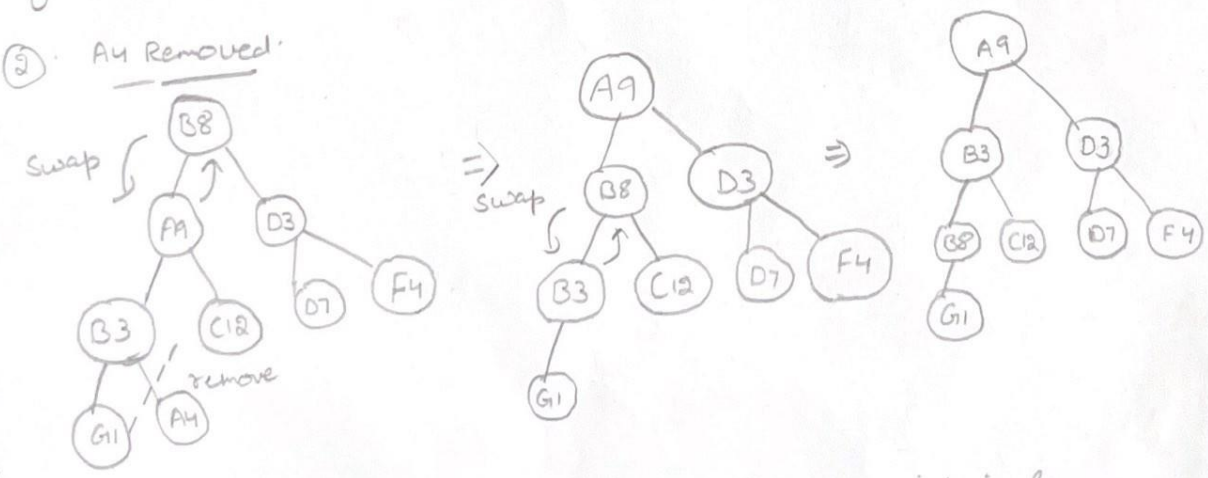
① Remove A2:





D7 gets swapped with A4, then with D3 because of minimum key A4- Heap order property.

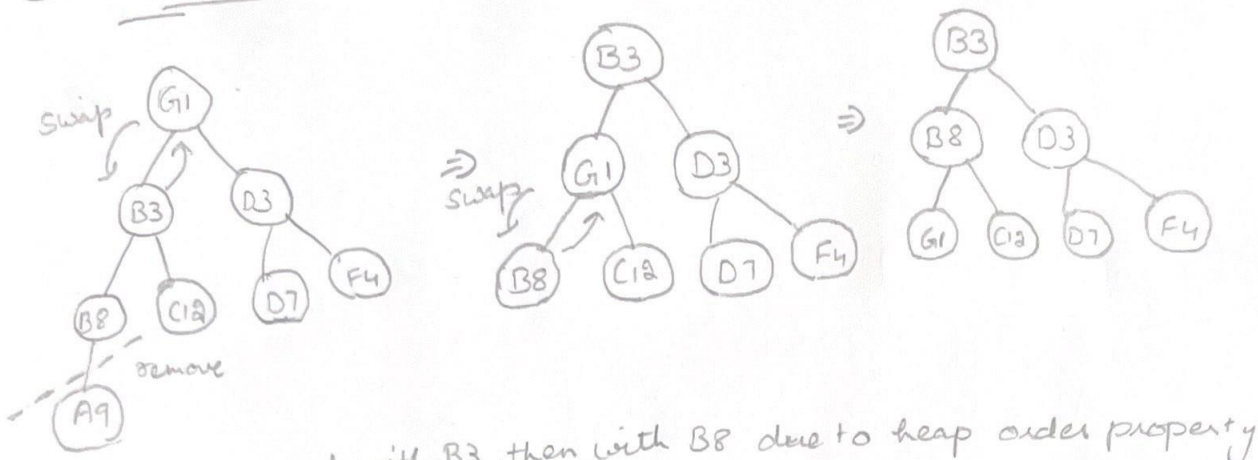
② A4 Removed.



B8 gets swapped with A9, then with B3 to maintain heap order property.

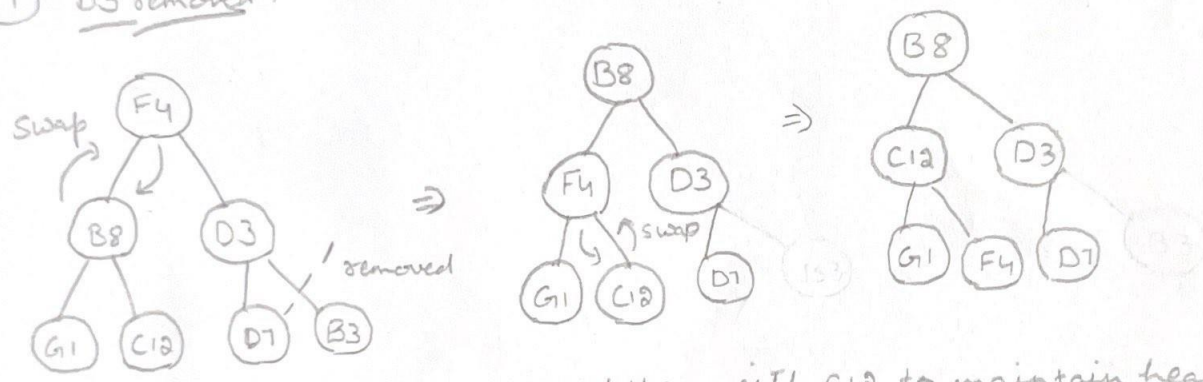
Default Comparator

(3) A9 Removed:



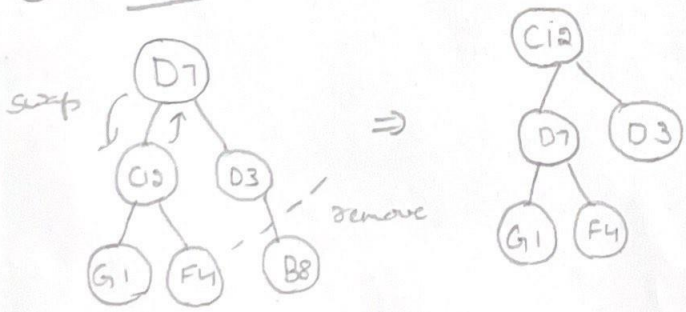
G1 gets swapped with B3, then with B8 due to heap order property since B3 has lowest key and becomes the root.

(4) B3 removed:



F4 gets swapped with B8 and then with C12 to maintain heap order property because F4 has biggest key among them.

(5) B8 removed:



D7 gets swapped with C12 due to heap order property; since C12 is the small key