

Instruction: You must show all your work clearly for credit. Partial credit will only be given to meaningful answers.

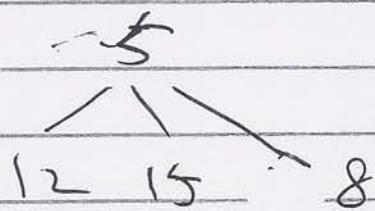
1. Given a set S of 15 records with priorities {5, 12, 15, 8, 9, 4, 1, 18, 6, 16, 11, 2, 16, 7, 8}. Perform the following operations. You must show your tree clearly after each insert/delete operations.
 - (a) (10) Build a min 3-heap for S by inserting the records, in the order given, into an initially empty heap. When done, perform deleteMin once followed by deleteMax once.
 - (b) (10) Build a max 3-heap for S by using the O(n) bottom-up buildheap operation. When done, perform deleteMin once followed by deleteMax once.
 - (c) (10) Build a minMax heap for S by inserting the records, in the order given, into an initially empty heap. When done, perform deleteMin once followed by deleteMax once.
 - (d) (10) Build a maxMin heap for S by using the O(n) bottom-up buildheap operation. When done, perform deleteMin once followed by deleteMax once.
 - (e) (10) Build a min leftist heap for S by inserting the records, in the order given, into an initially empty heap. When done, perform deleteMin once. Illustrate the data structure of the final heap.
 - (f) (10) Build a max leftist heap for S by inserting the records, in the order given, into an initially empty heap. When done, perform deleteMax once. Illustrate the data structure of the final heap.
 - (g) (10) Build a min skew heap for S by inserting the records, in the order given, into an initially empty heap. When done, perform deleteMin once. Illustrate the data structure of the final heap.
 - (h) (10) Build a max skew heap for S by inserting the records, in the order given, into an initially empty heap. When done, perform deleteMax once. Illustrate the data structure of the final heap.
 - (i) (10) Build a min pairing heap for S by inserting the records, in the order given, into an initially empty heap. When done, perform deleteMin once and use the two-pass method for merging. Show the data structure of the final heap.
 - (j) (10) Build a max pairing heap for S by inserting the records, in the order given, into an initially empty heap. When done, perform deleteMax once and use the multi-ass method for merging. Show the data structure of the final heap.

Remark: To resolve ambiguity, you must operate from left (preferred) to right.

Hw #4

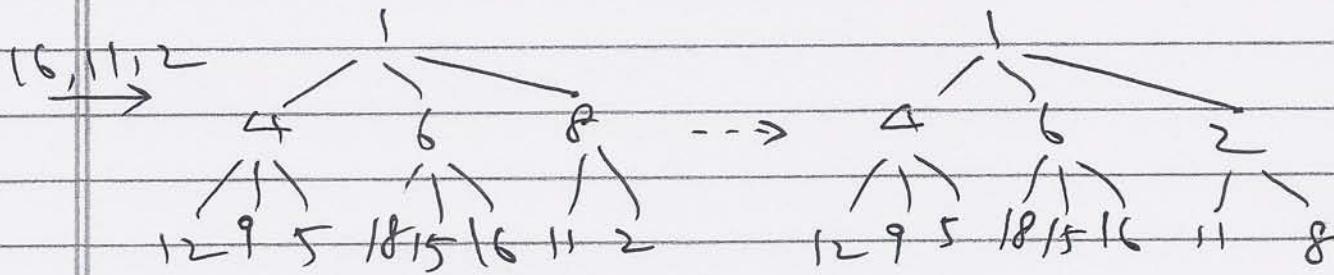
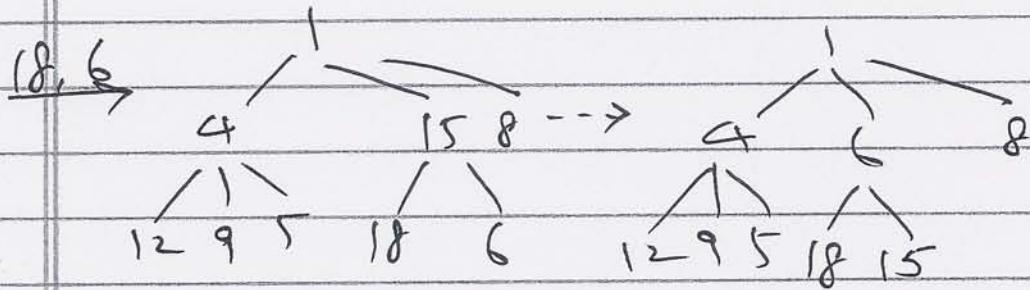
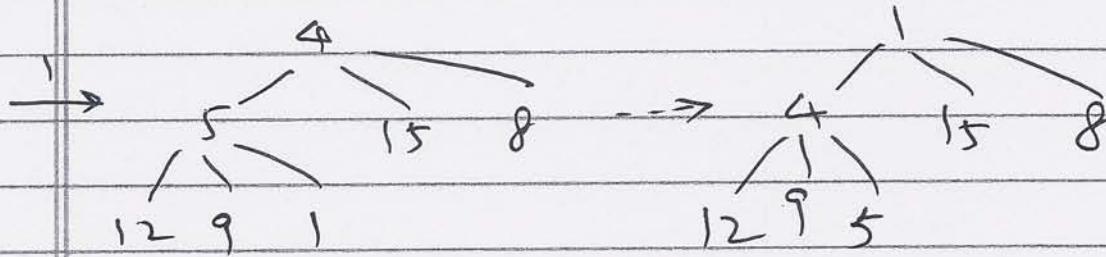
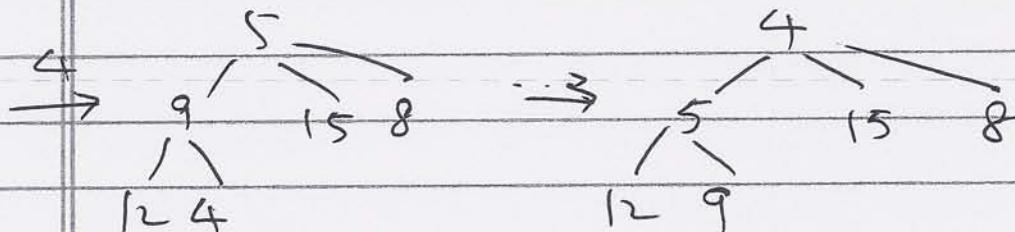
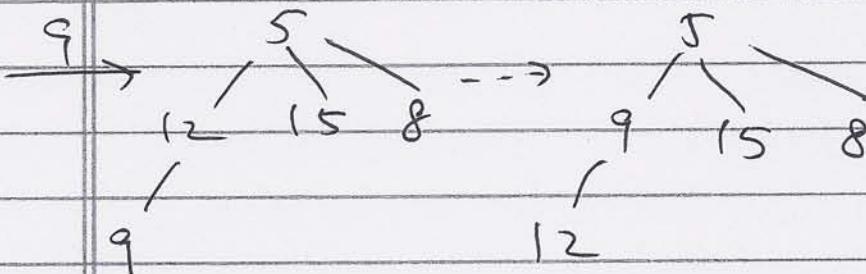
(a)

5, 12, 15, 8

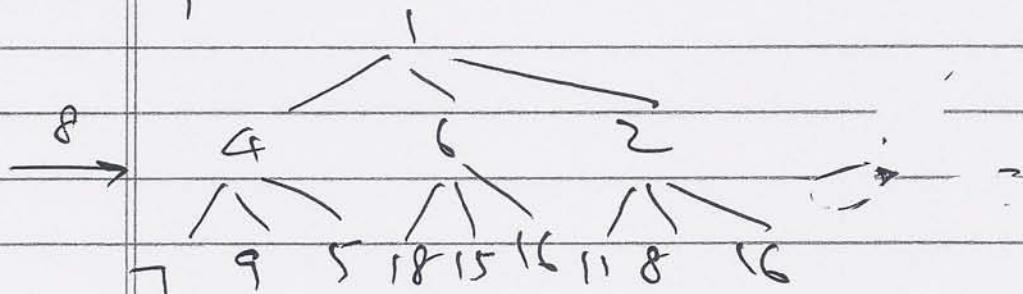
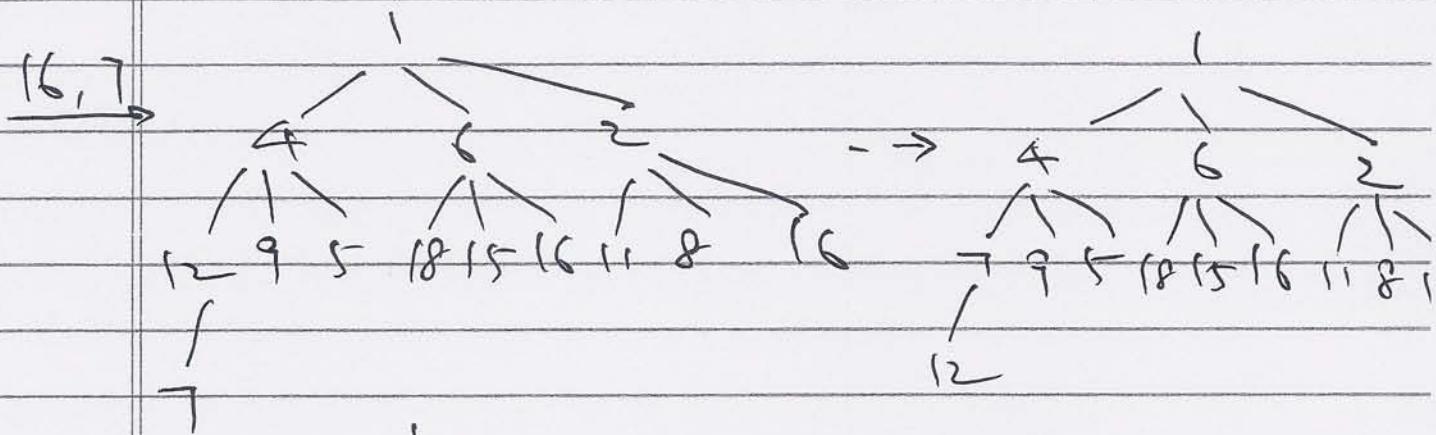


1.

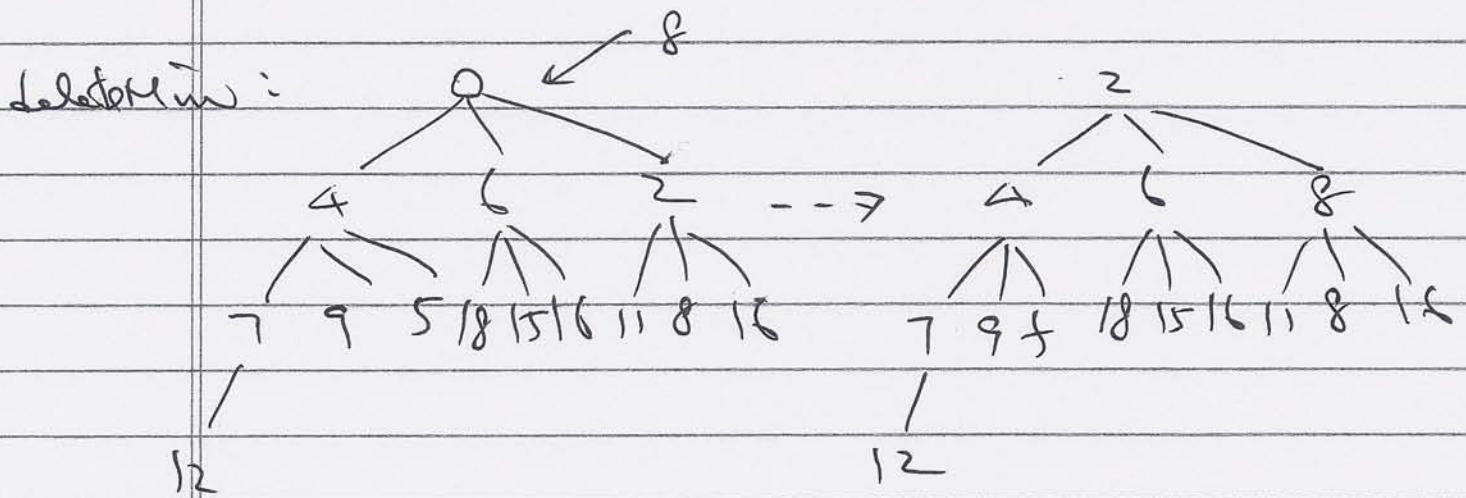
min-heap



(2)



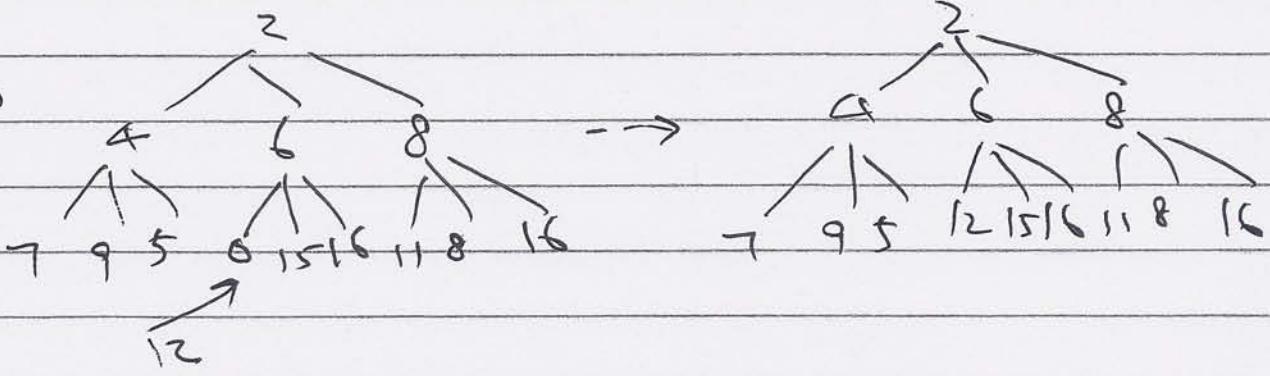
12 8



12

12

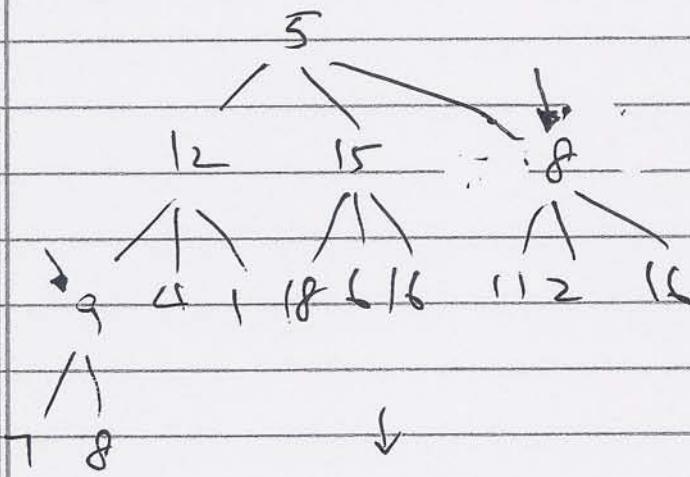
deleteMax



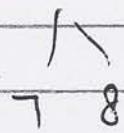
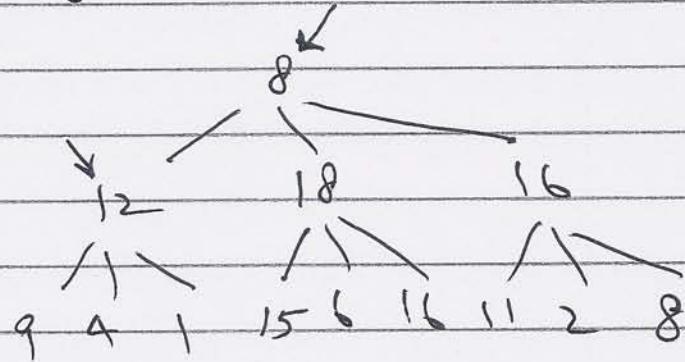
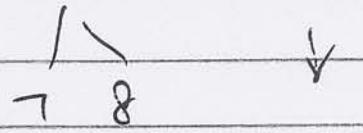
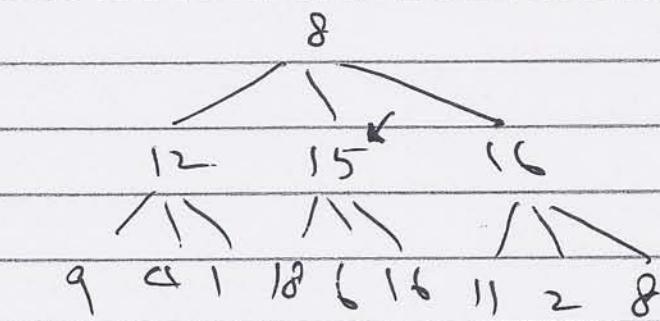
12

(6)

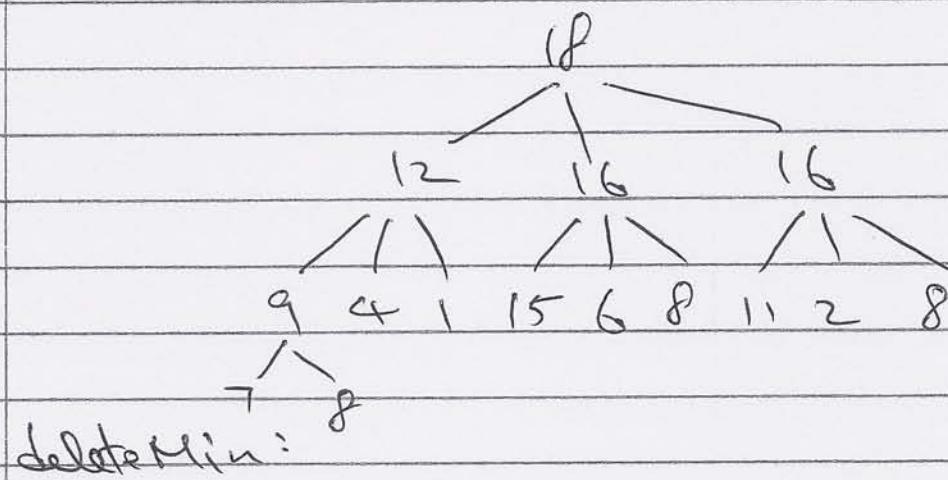
Complete 3-tree:



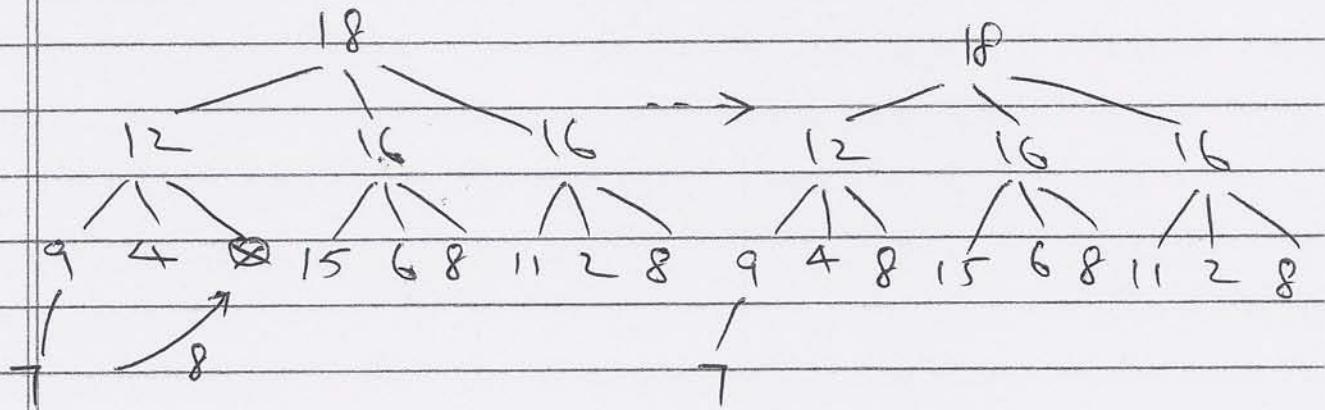
Max 3-heap



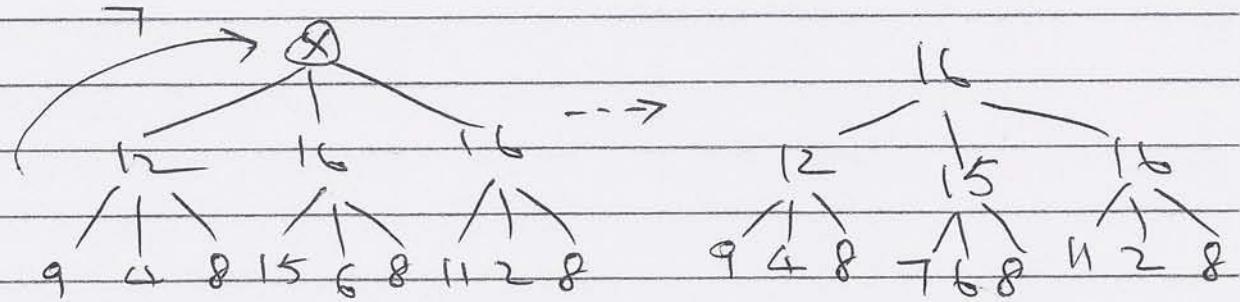
(4)



deleteMin:



deleteMax:

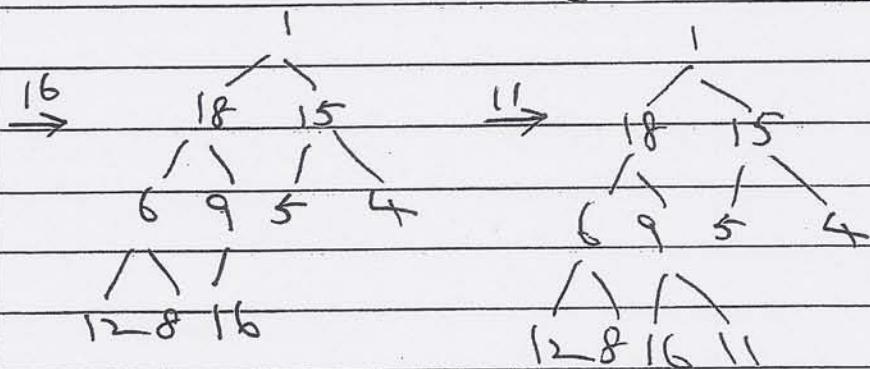
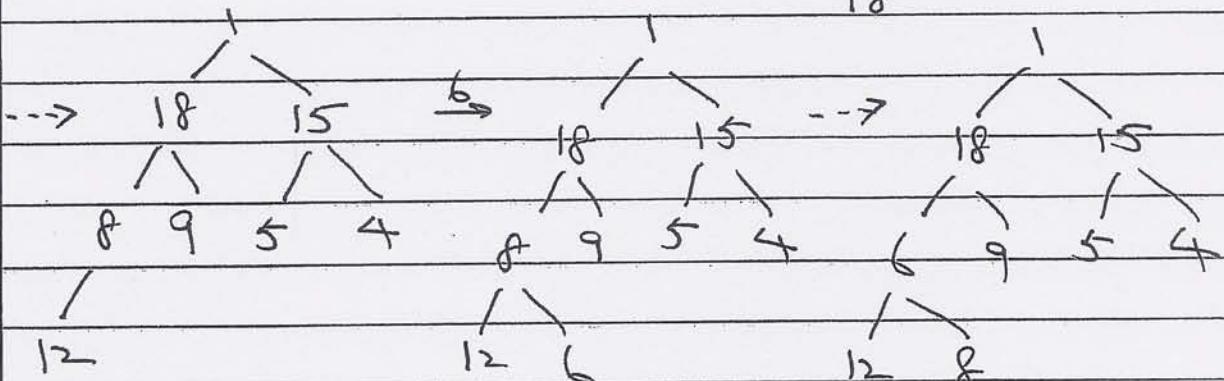
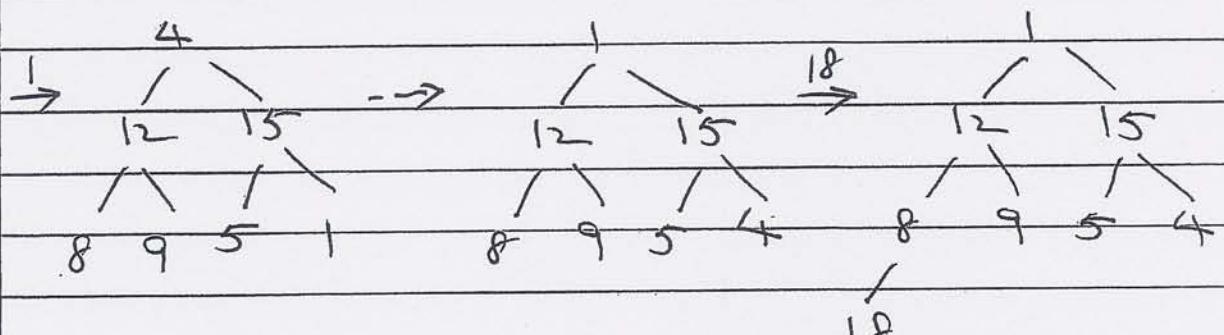
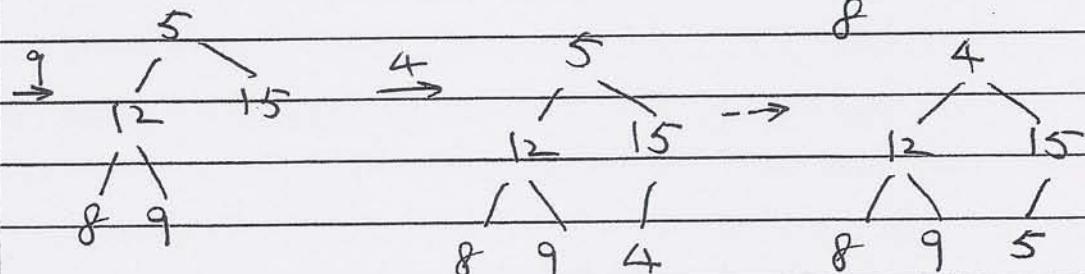
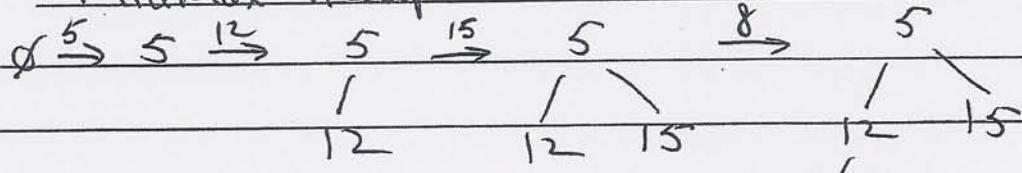


left to right

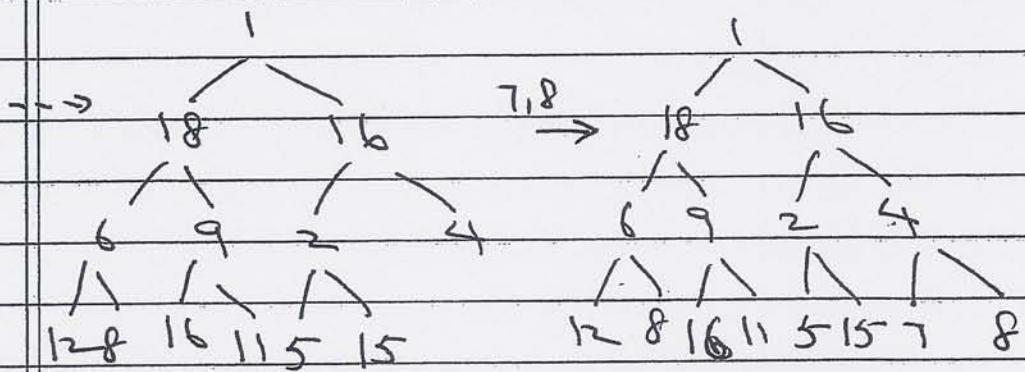
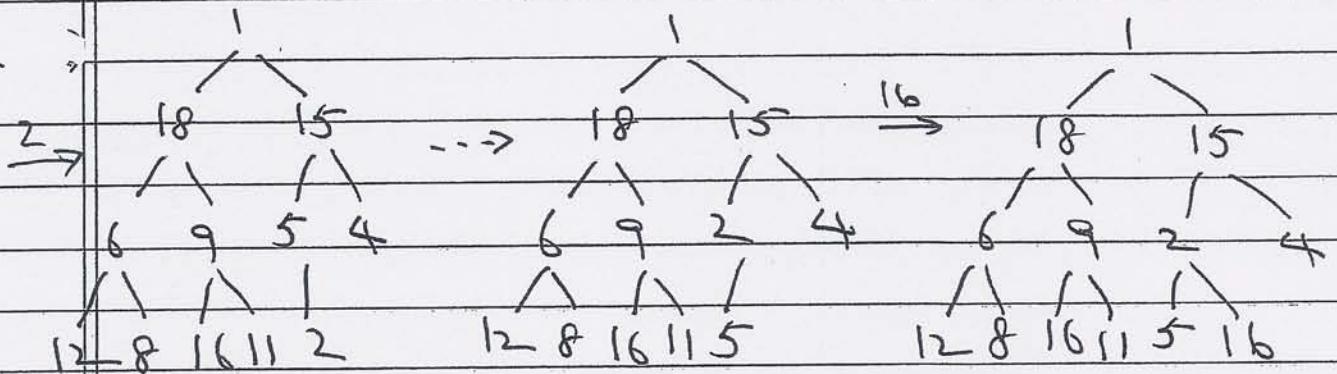
(d)

(c)

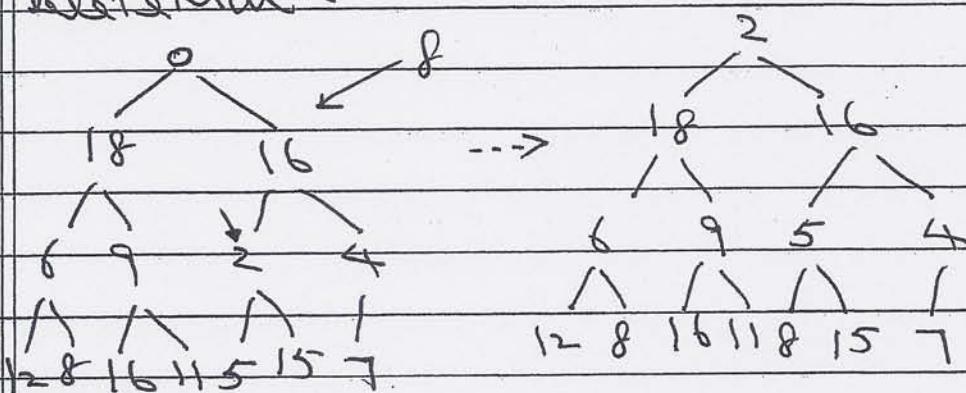
MinMax Heap:



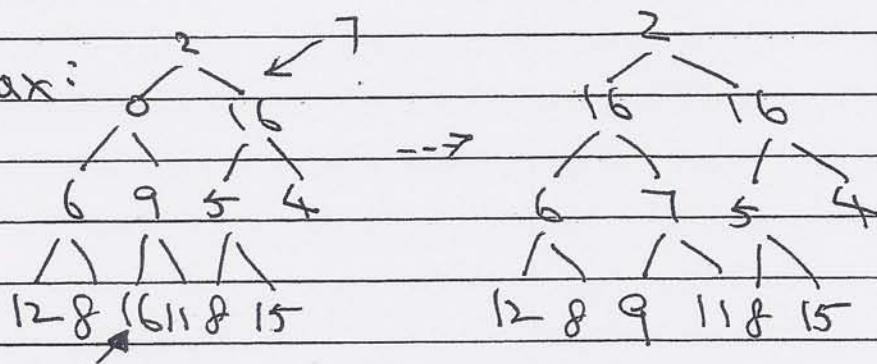
(6)



DeleteMin:

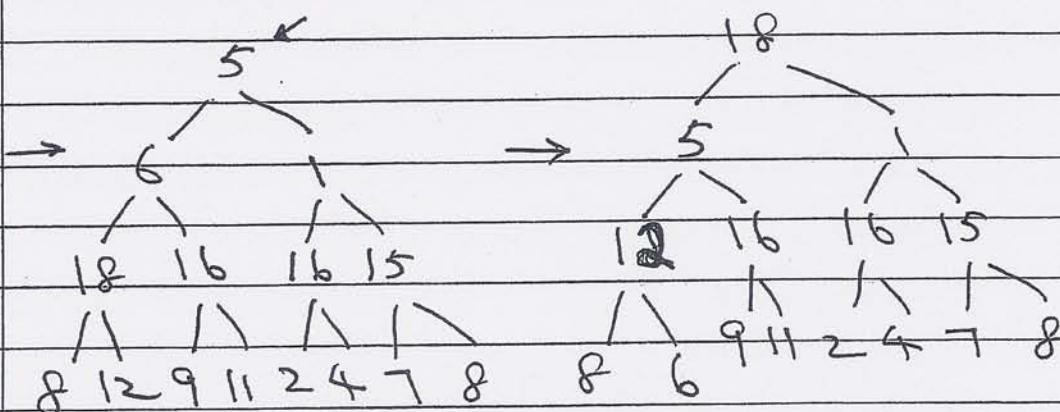
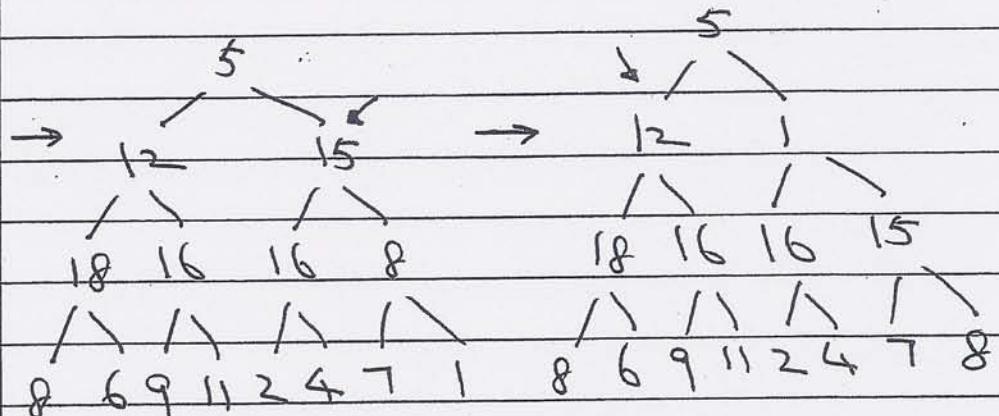
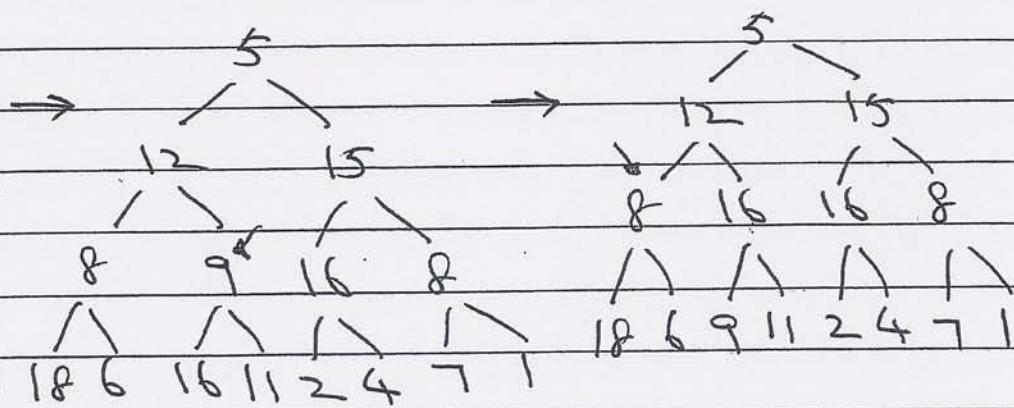
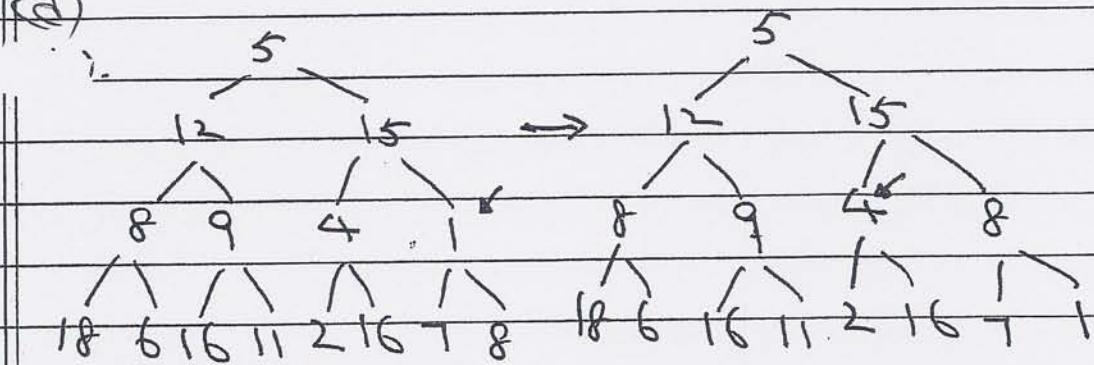


DeleteMax:



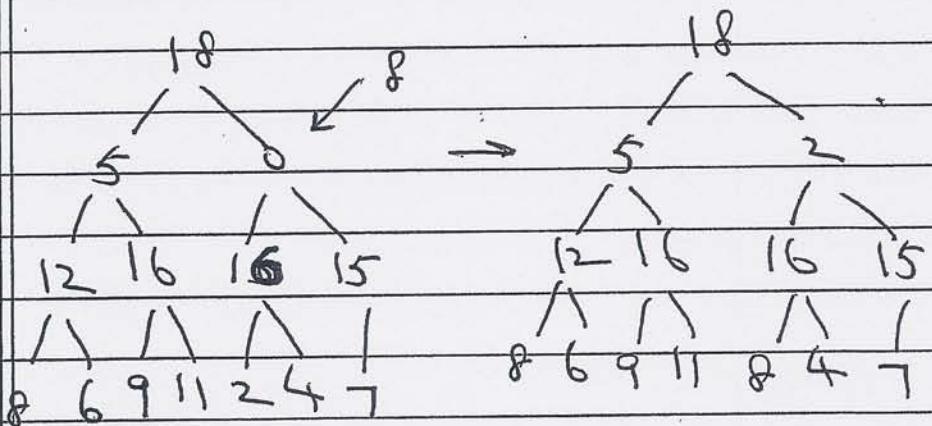
Max Min Heap:

7

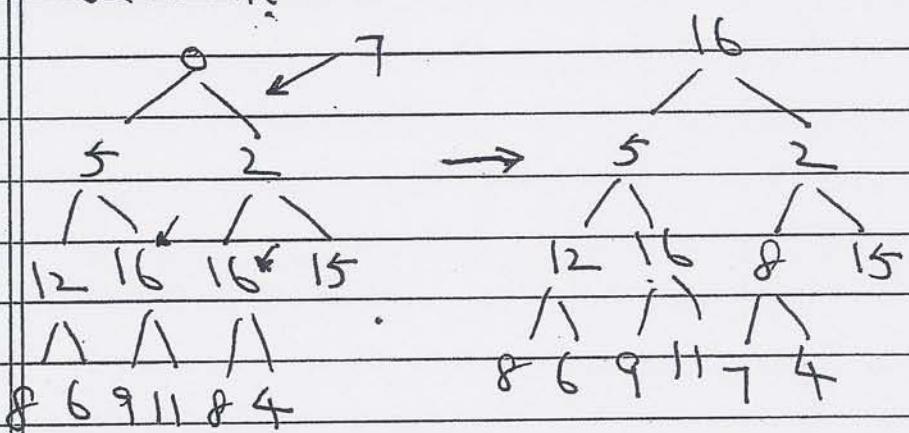


8

Delete Min

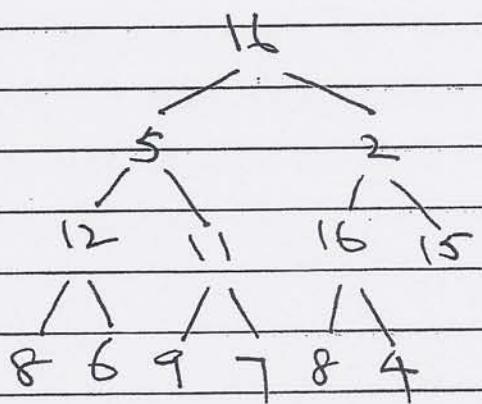


Delete Max:



= using right 16

↓ or

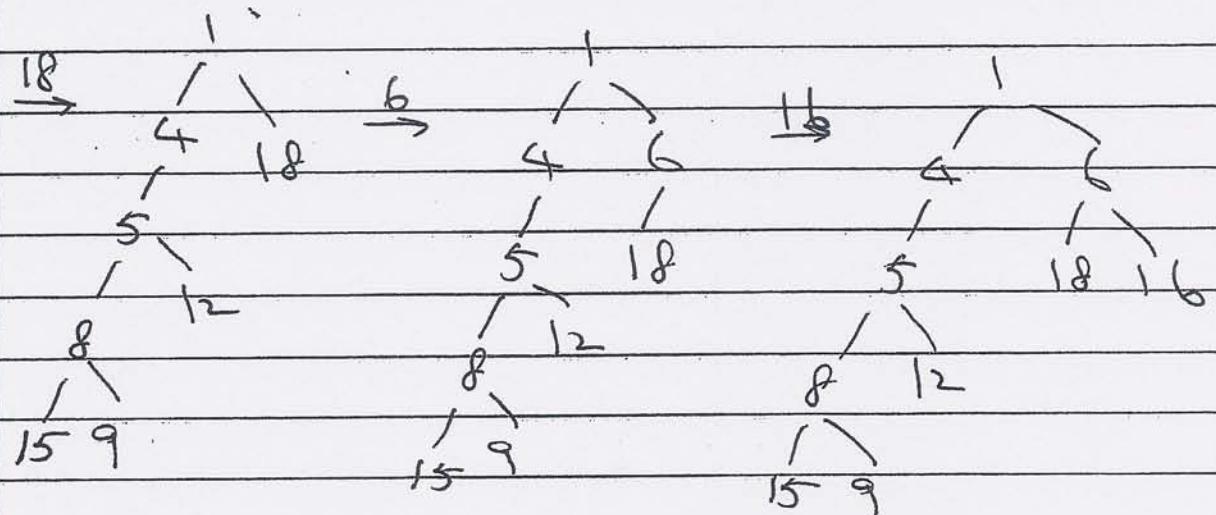
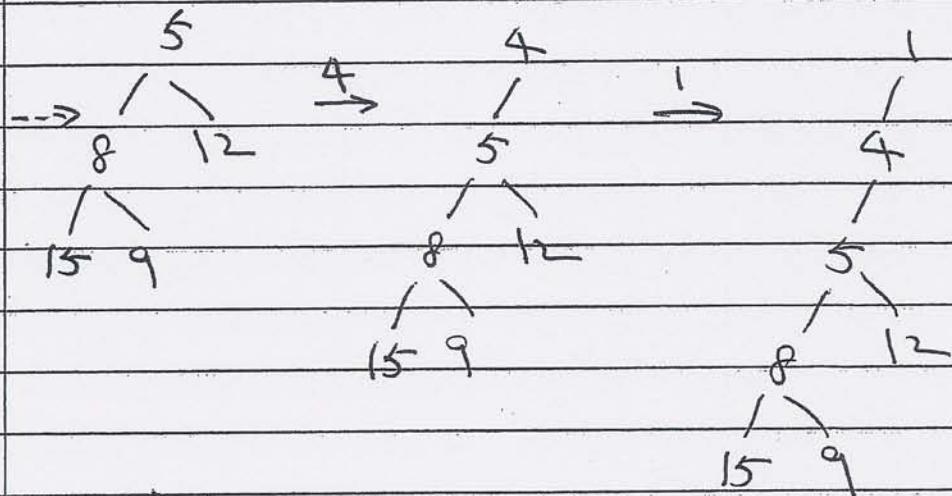
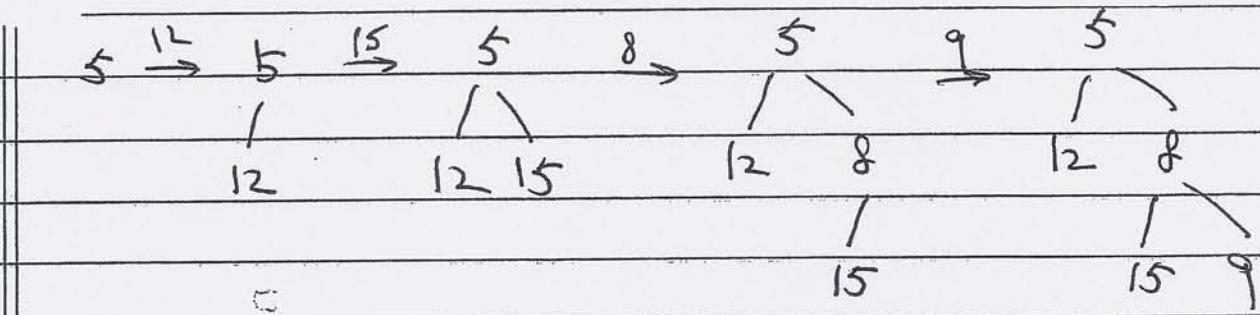


= using left 16

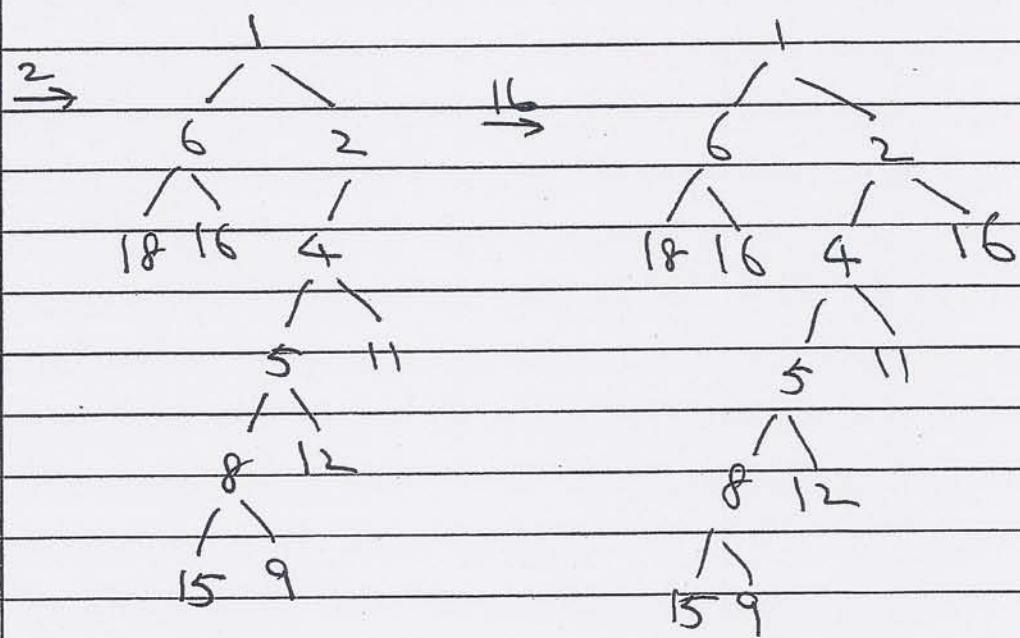
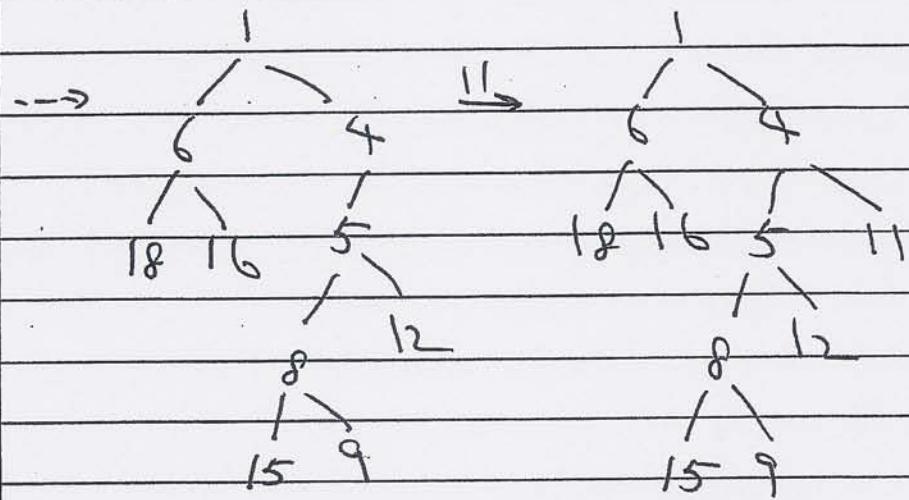
Min leftist Heap:

(2)

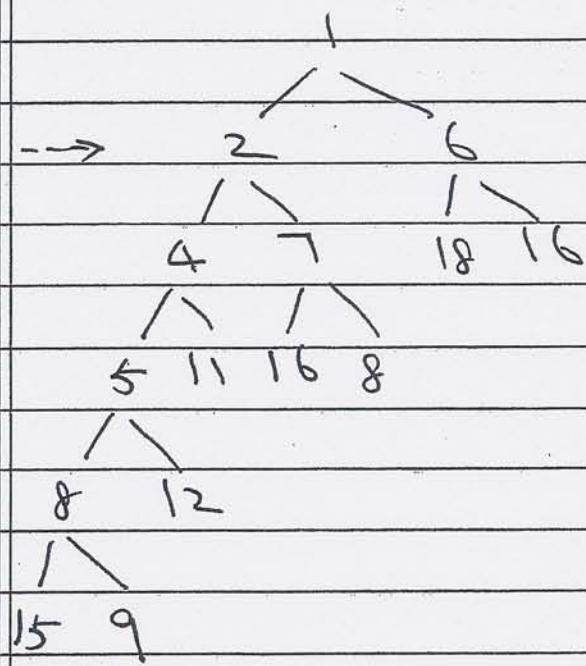
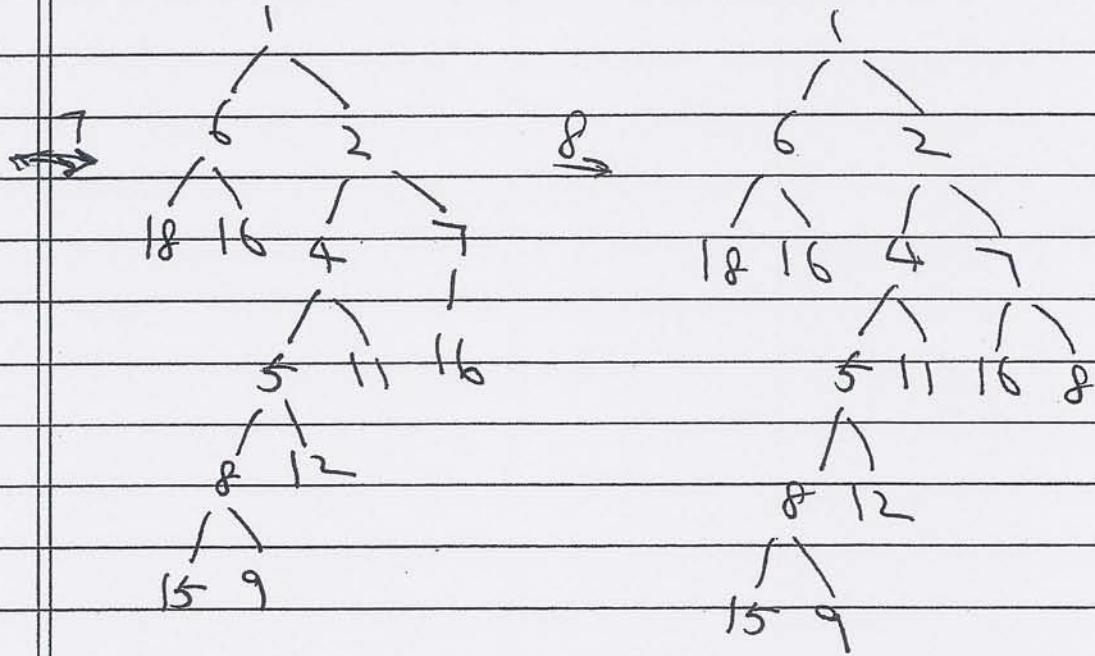
9



(10)

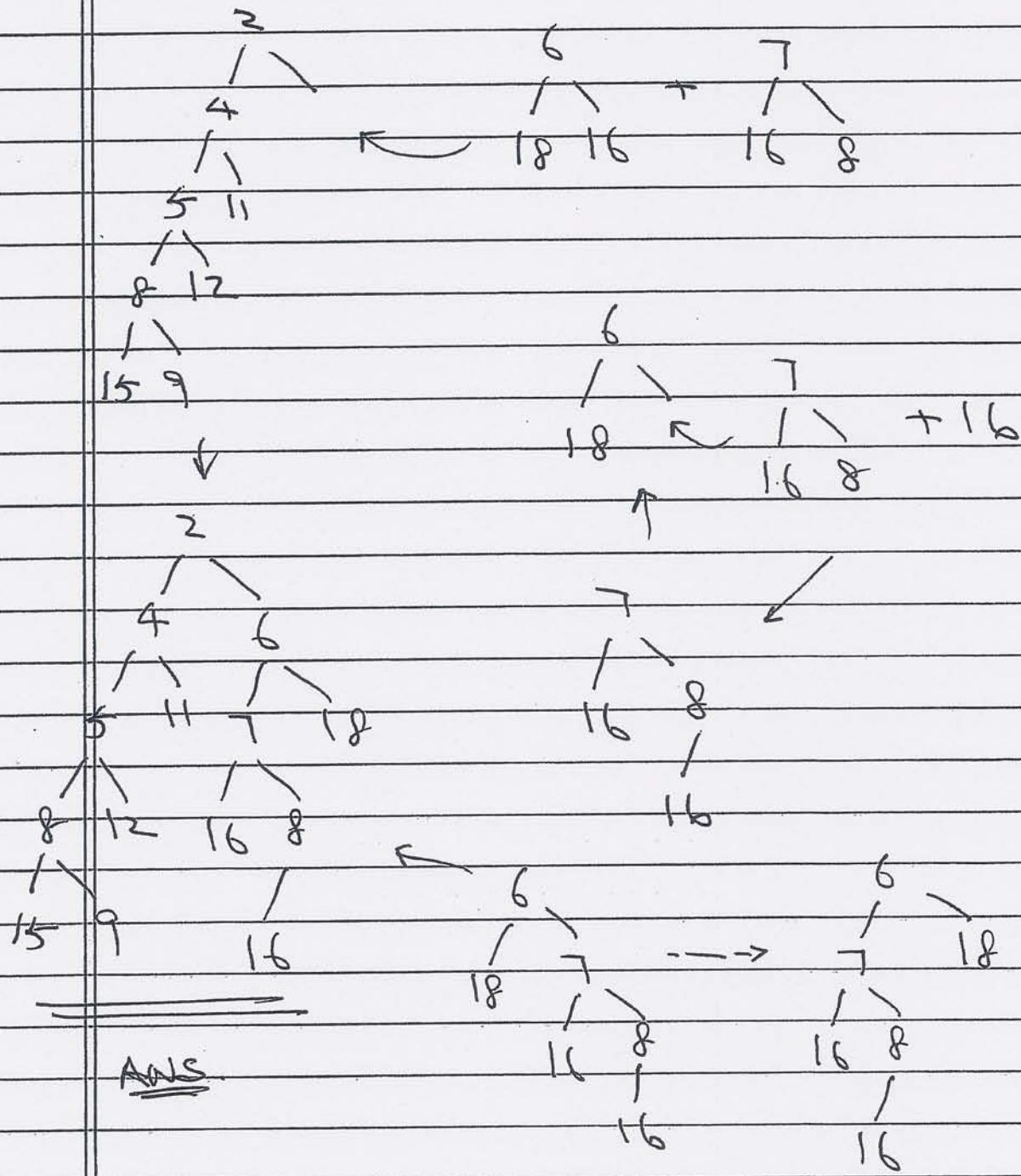


(11)



(12)

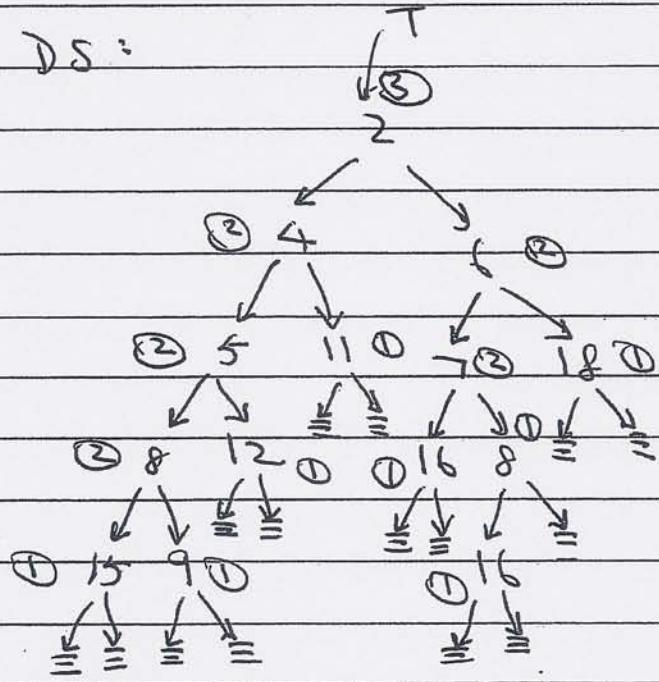
Delete Min:



Ans

13

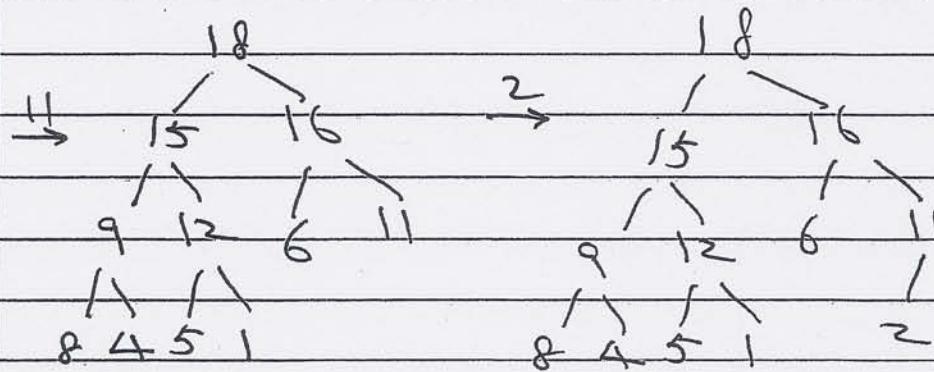
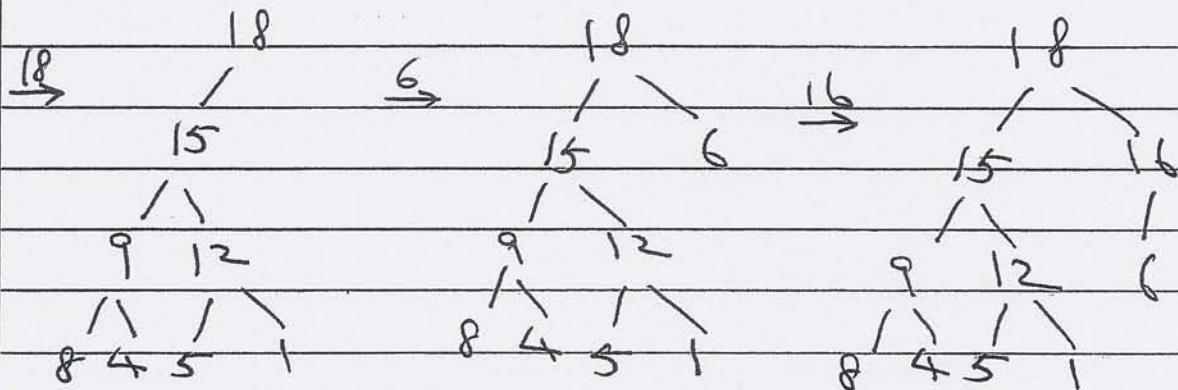
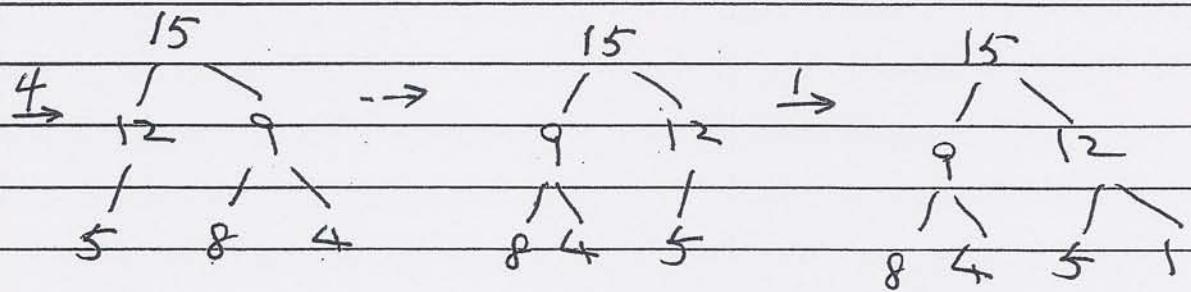
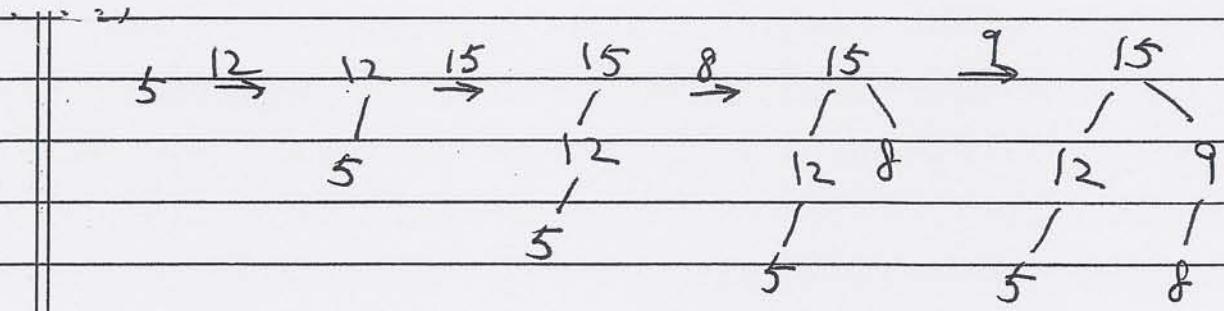
→ DS:



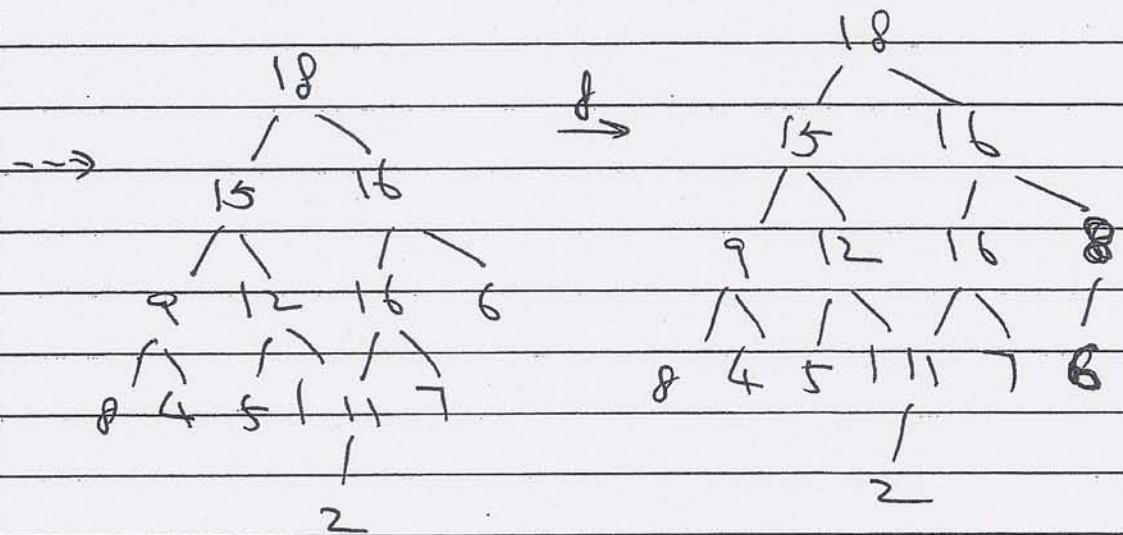
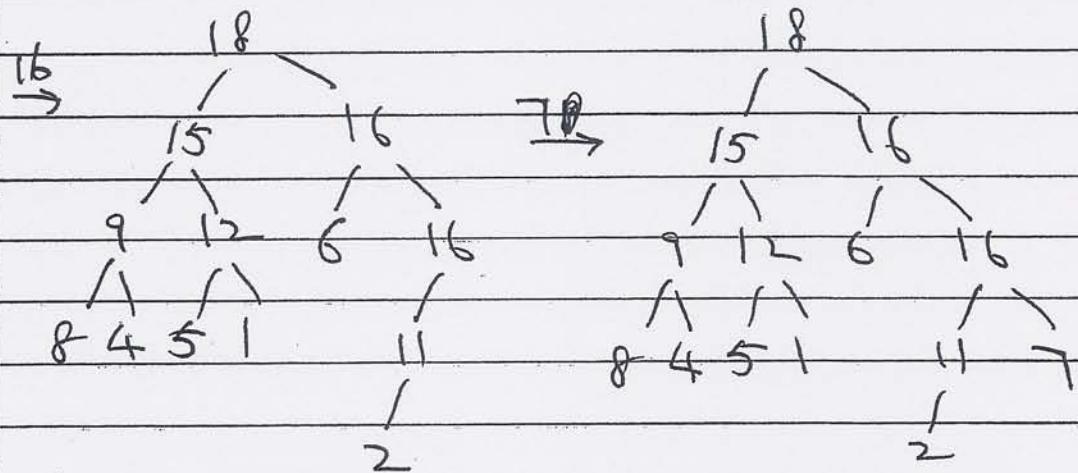
Must show rank of nodes!

Max Leftist Heap:
(f)

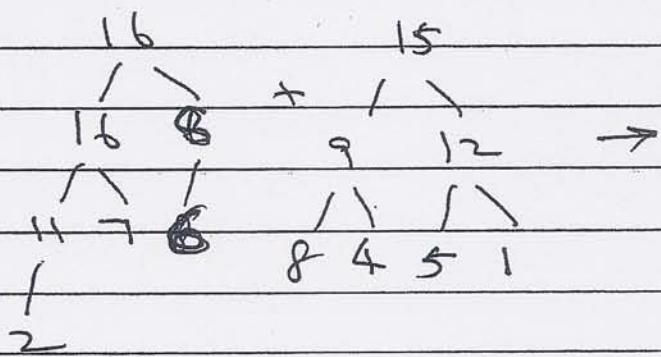
14



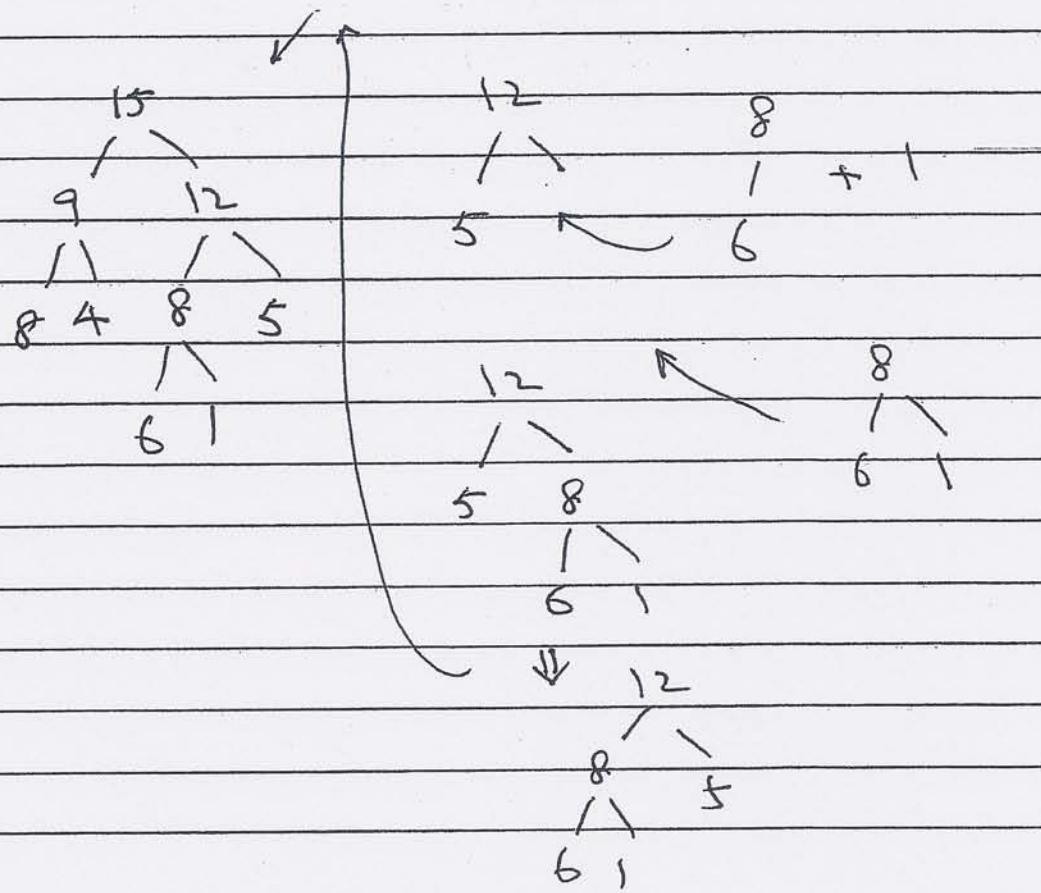
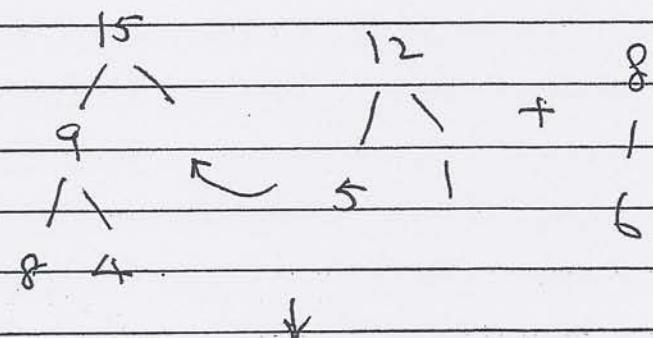
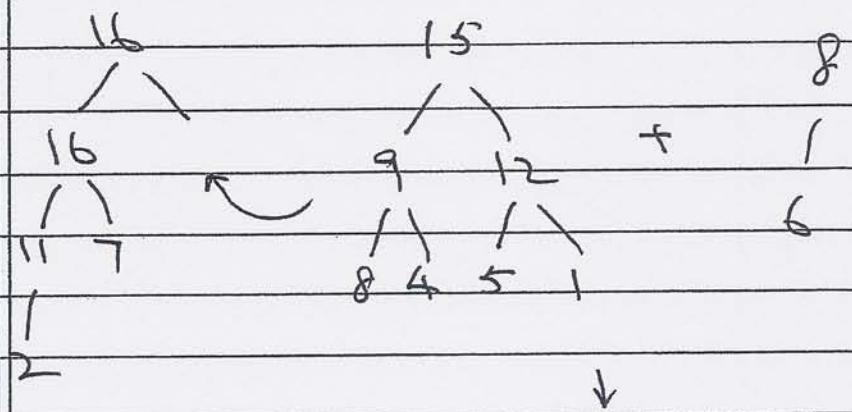
(15)



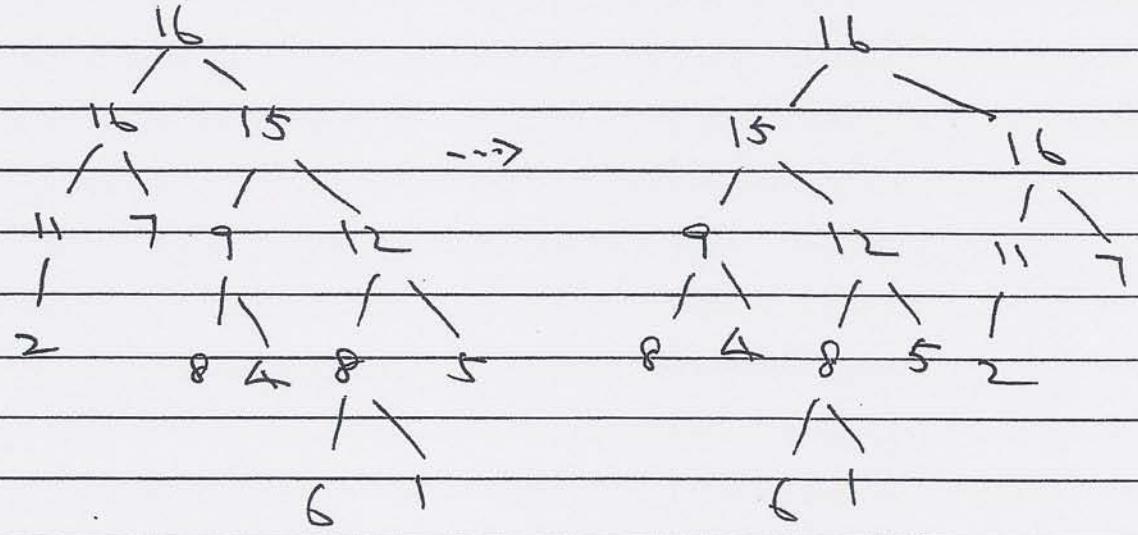
Deletion:



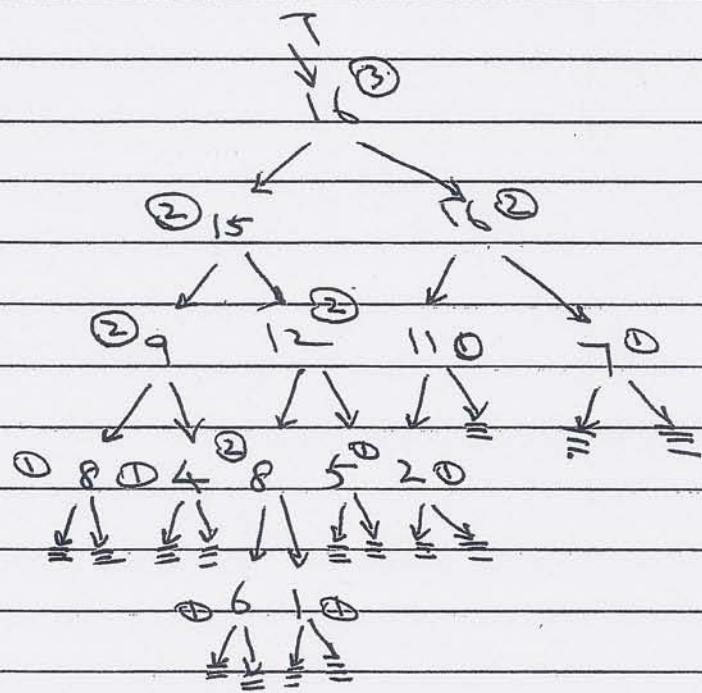
(16)



(17)



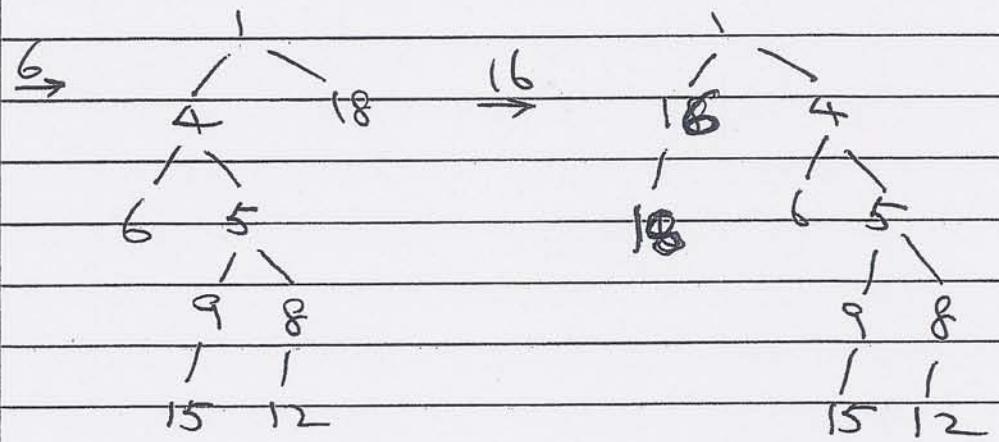
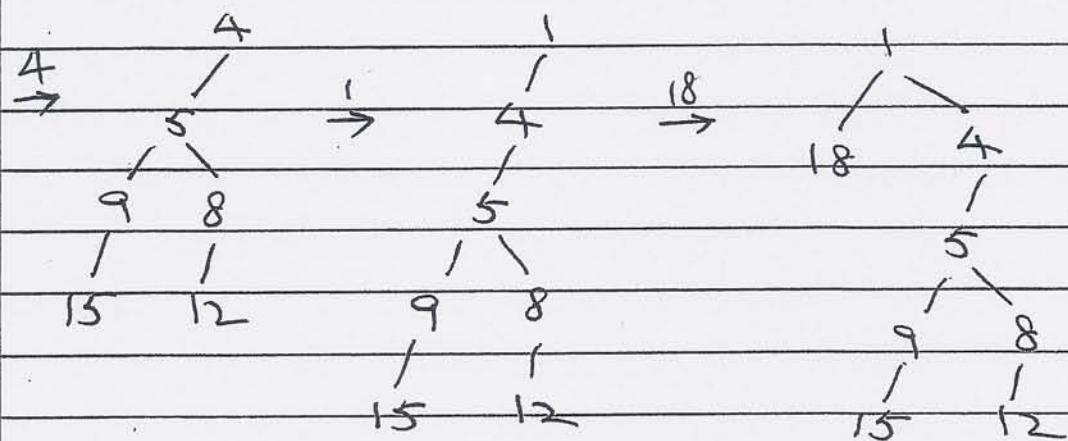
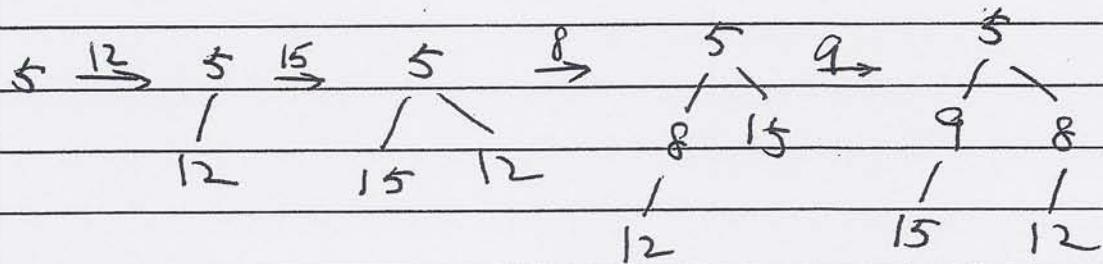
08:



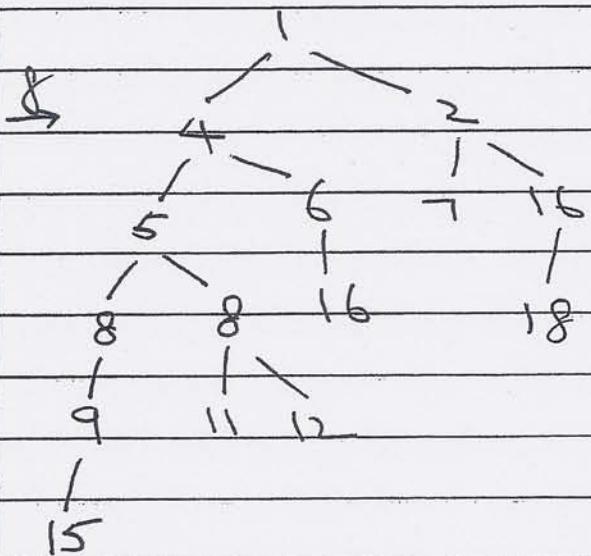
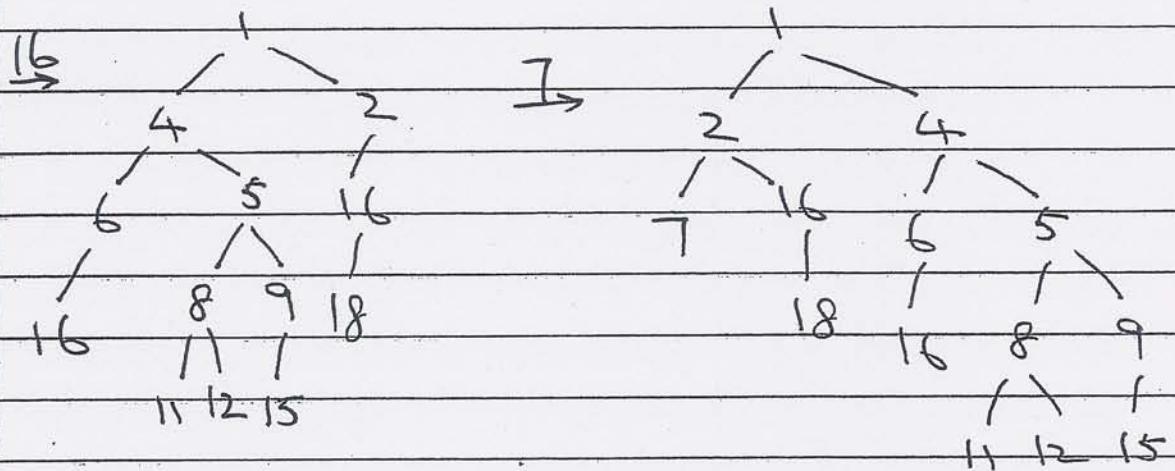
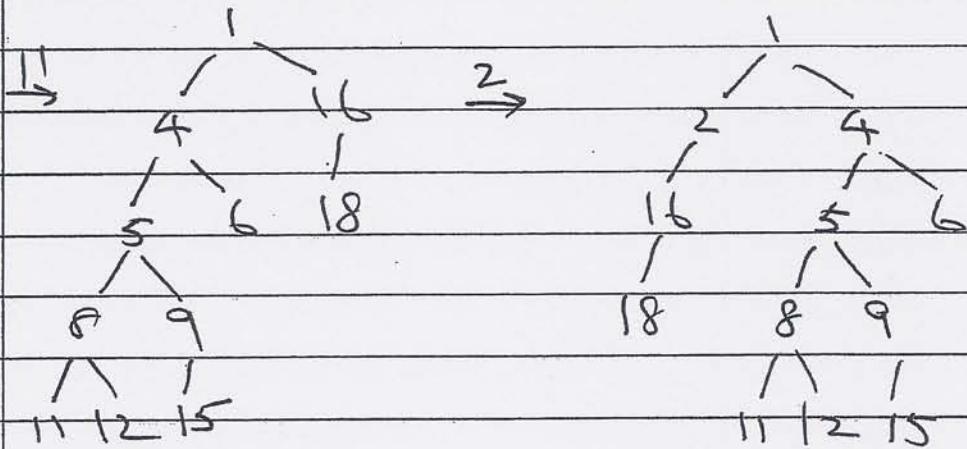
(g)

18

Min-Heap

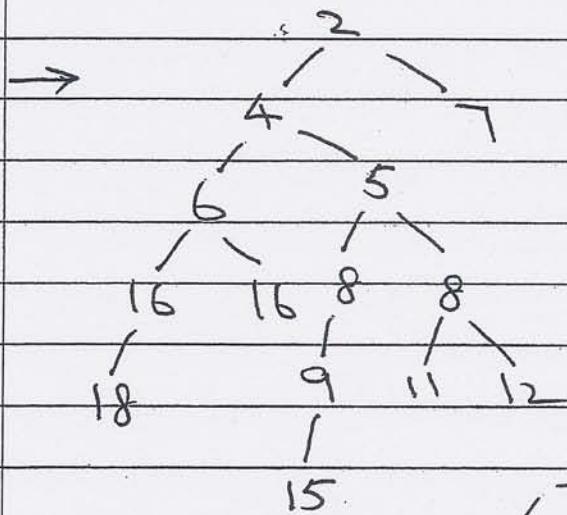
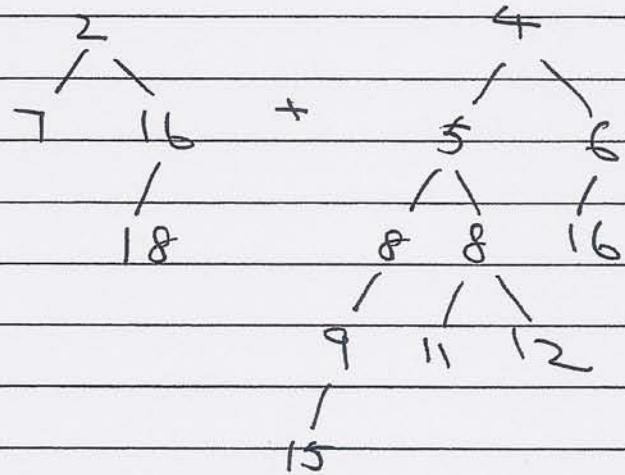


(19)

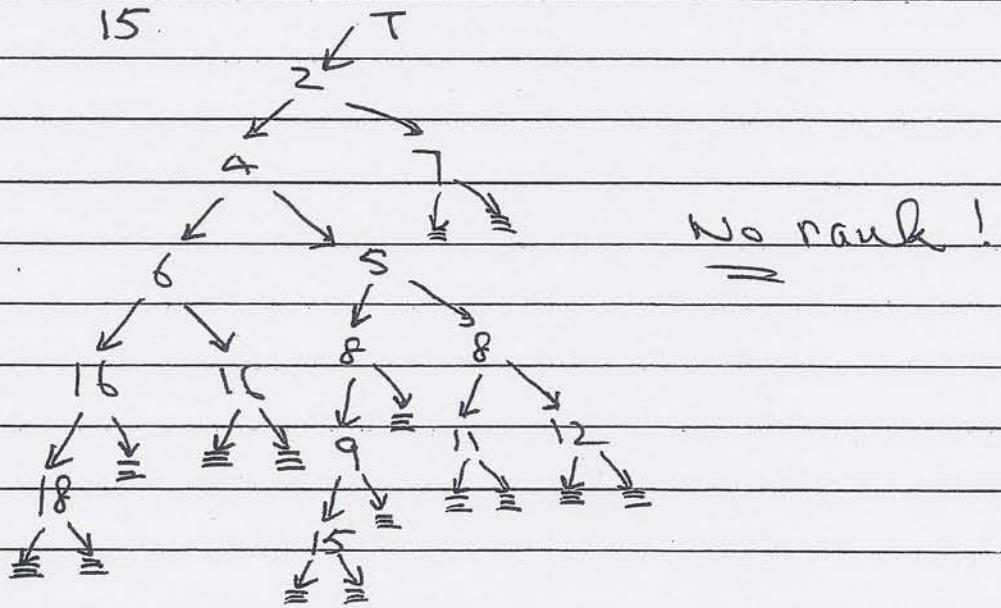


(2)

Determinant:

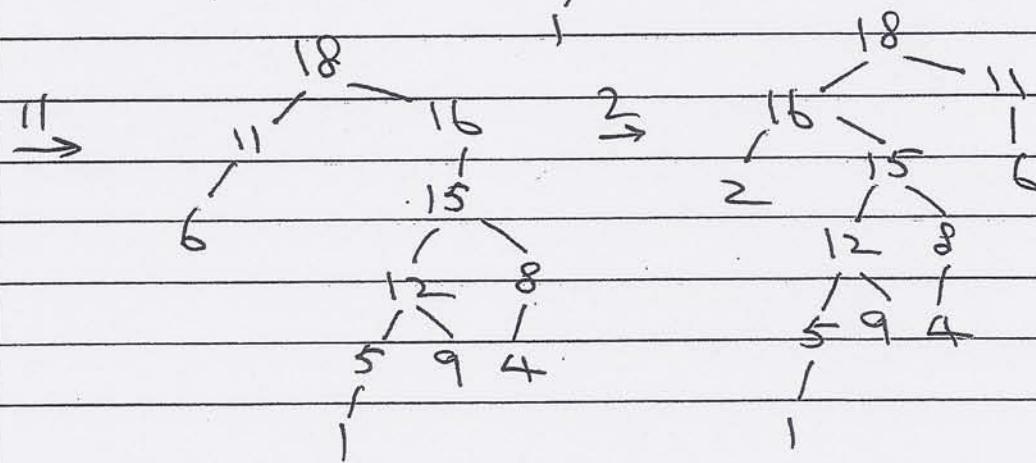
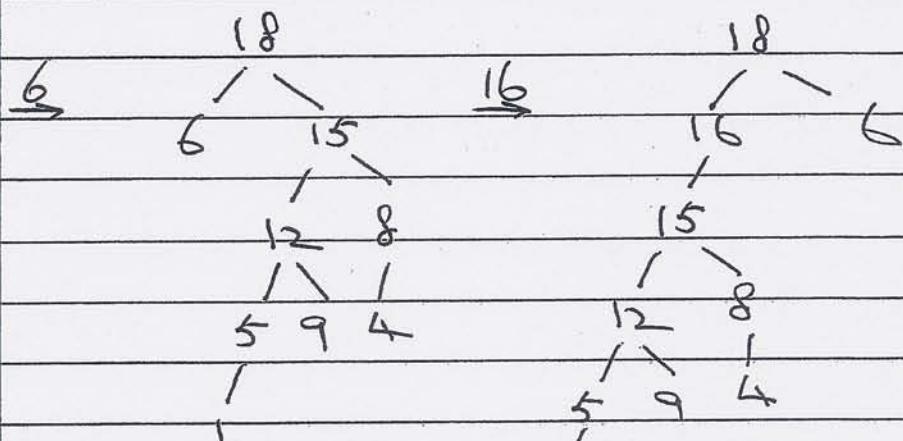
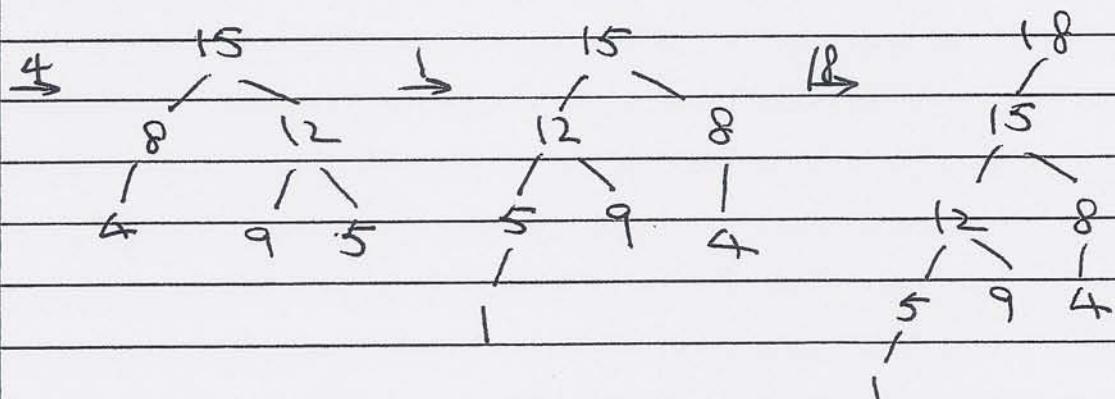
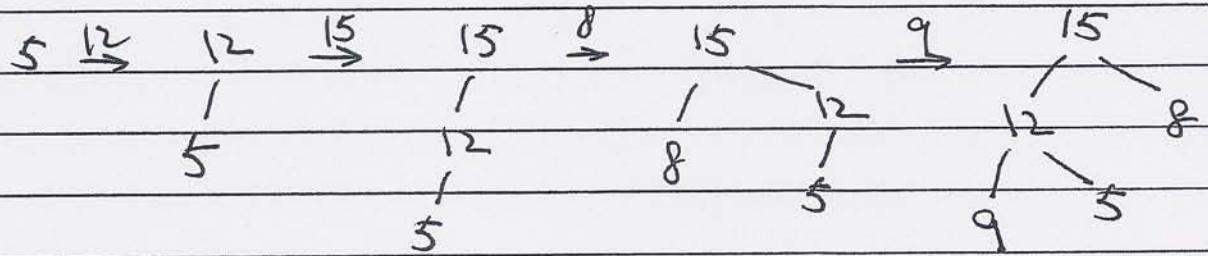


DS:

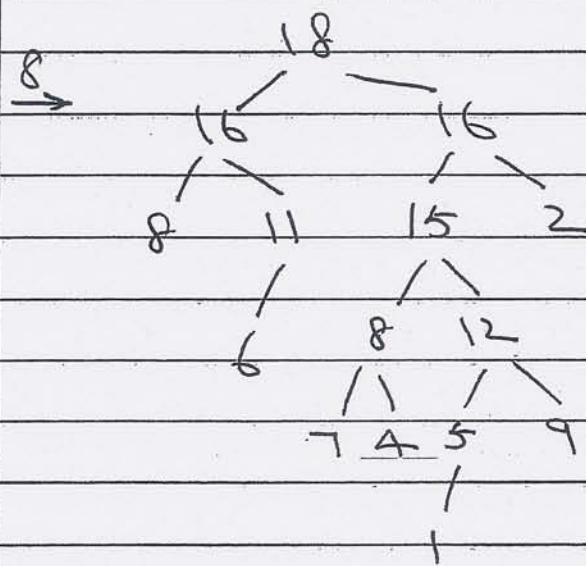
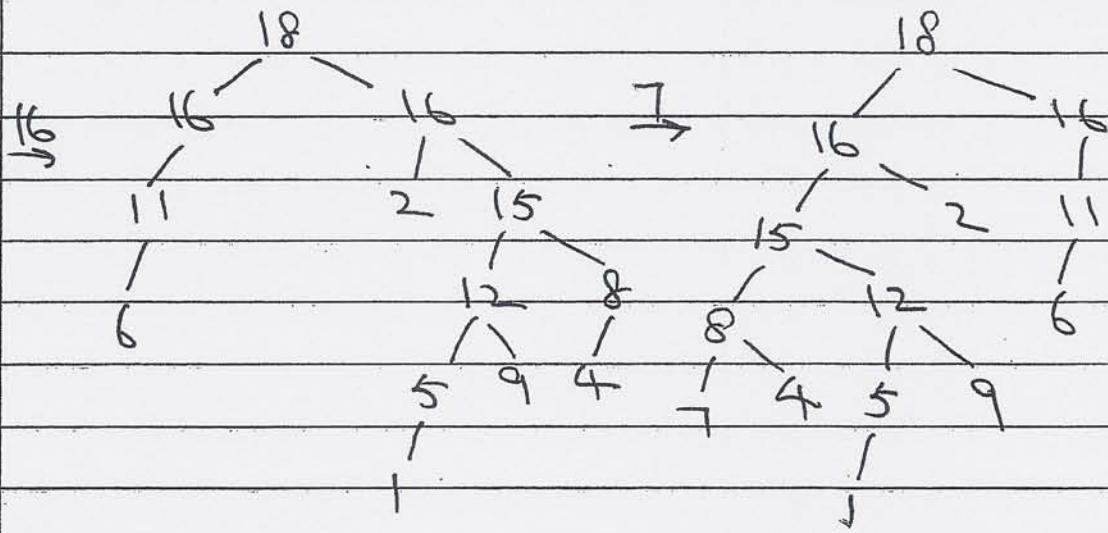


(4)

(21)

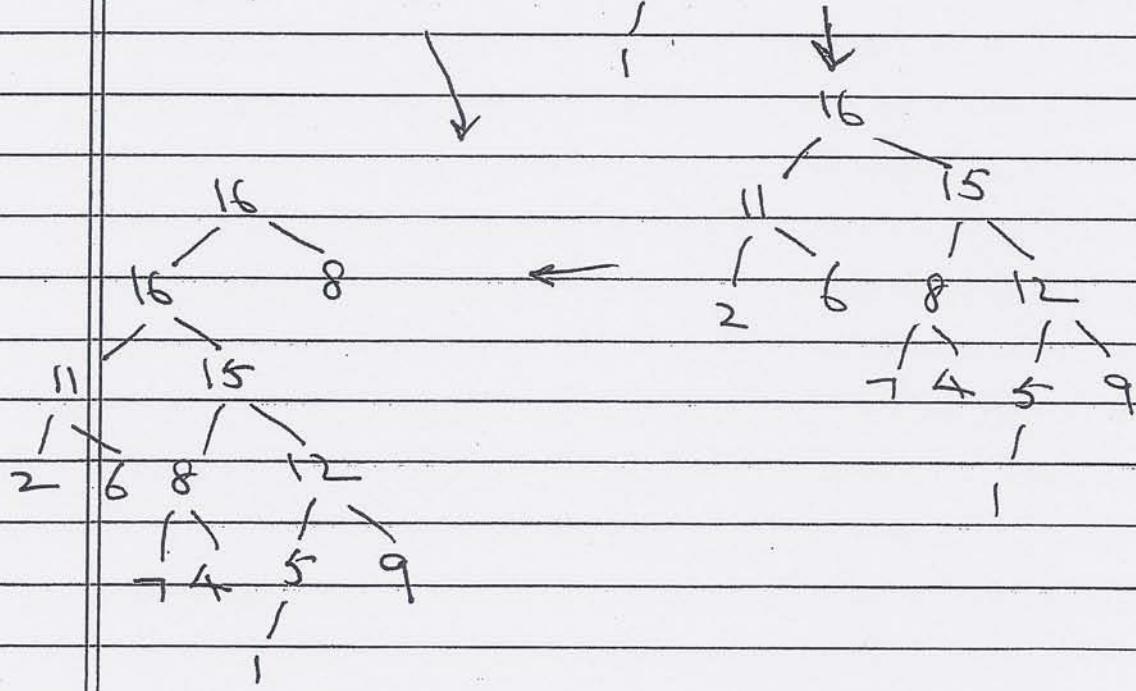
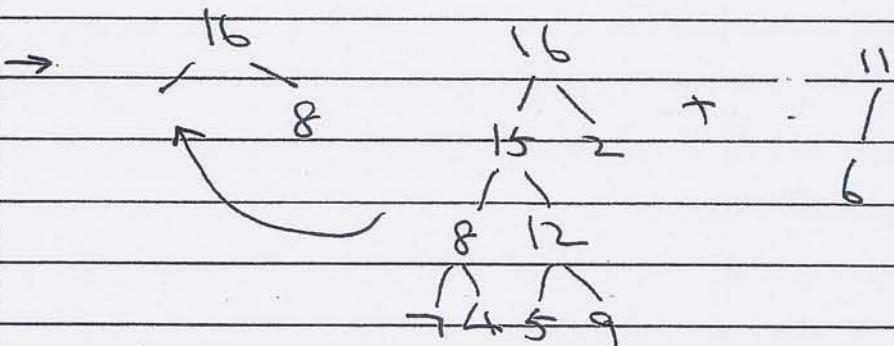
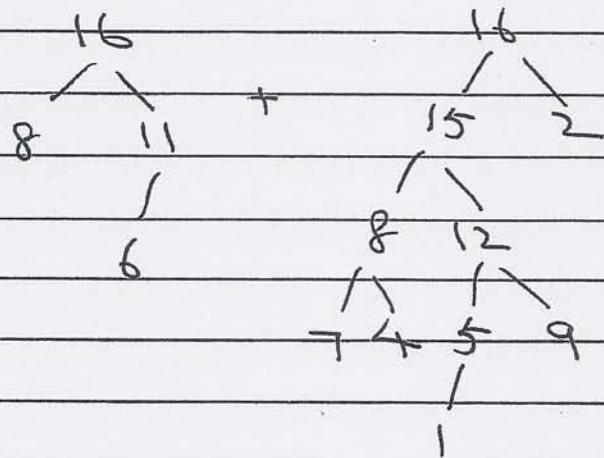
Max Skew Heap:

(22)



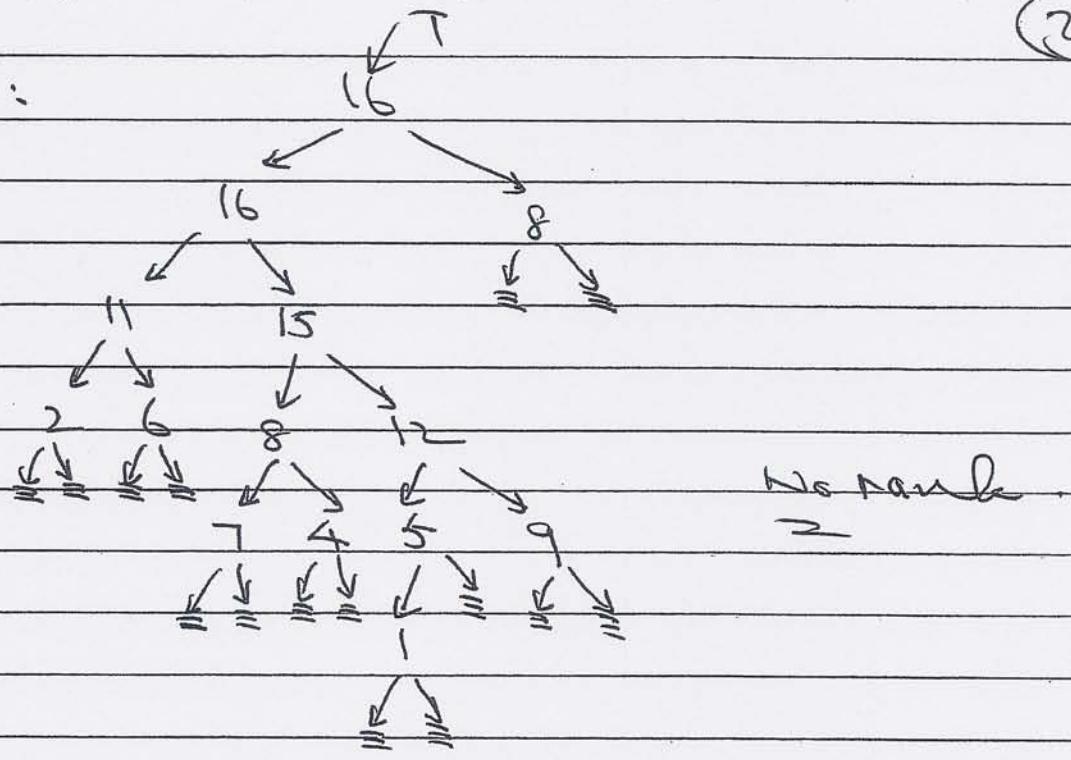
(23)

Delete Max:



24

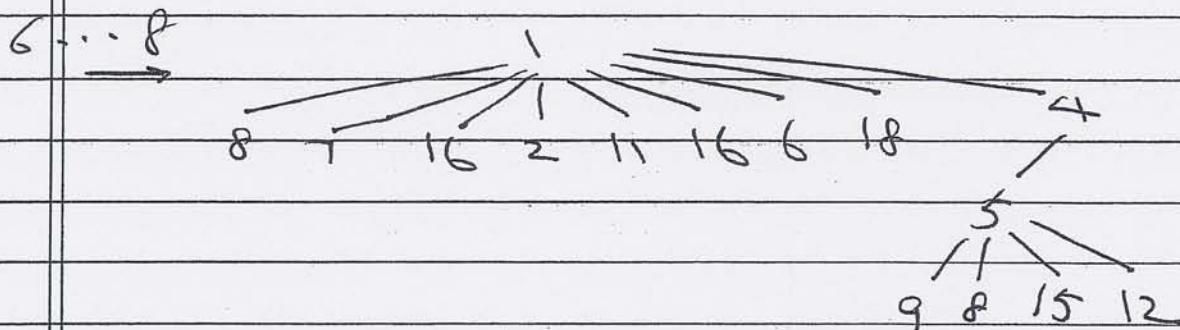
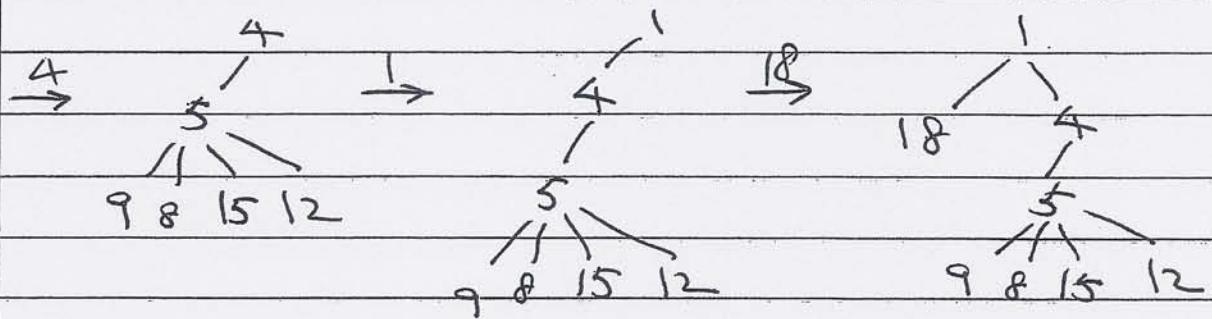
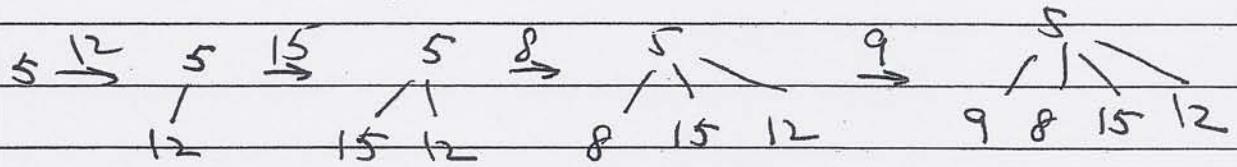
DS:



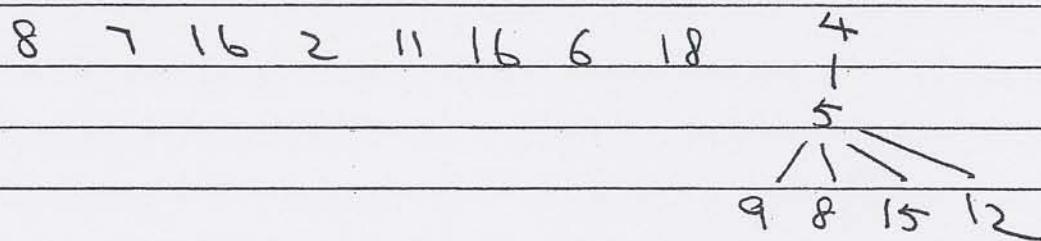
(i)

25

Min Pairing Heap:

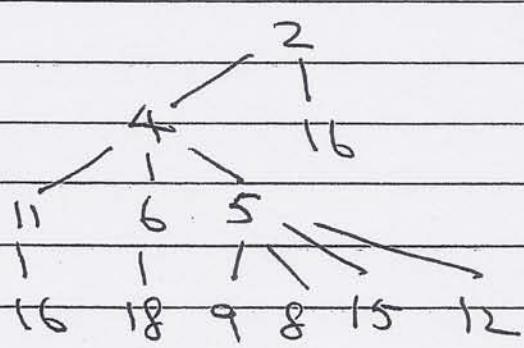
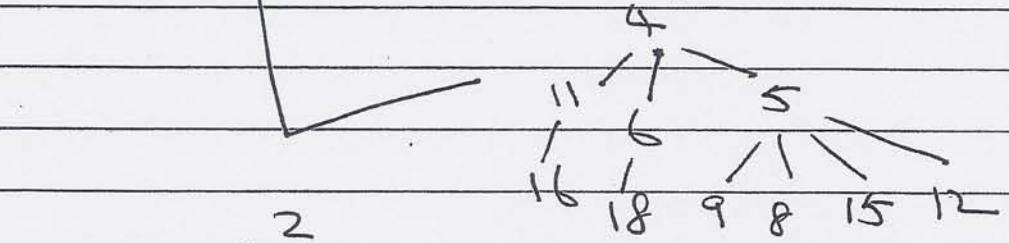
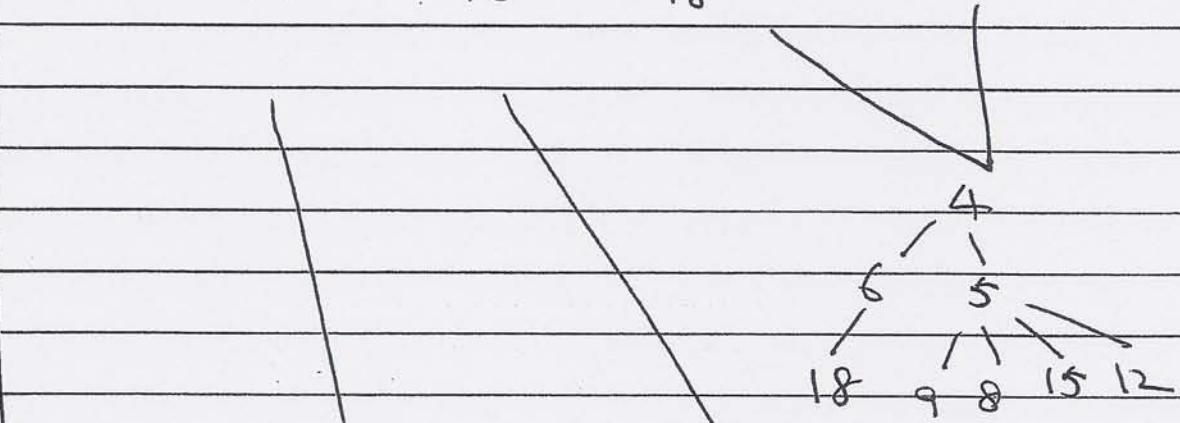
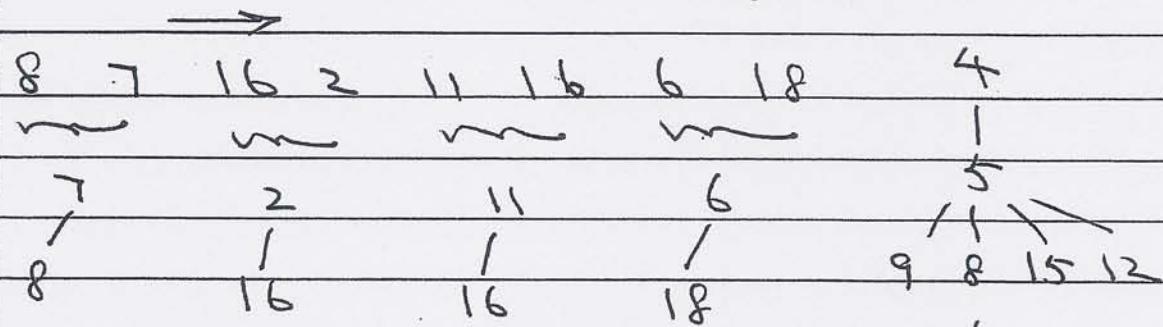


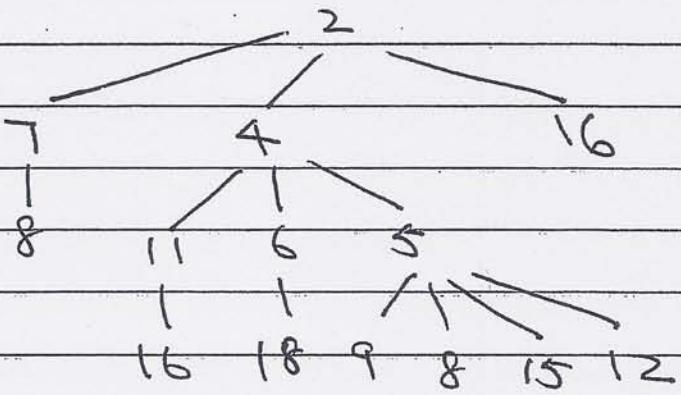
Delete min:



(26)

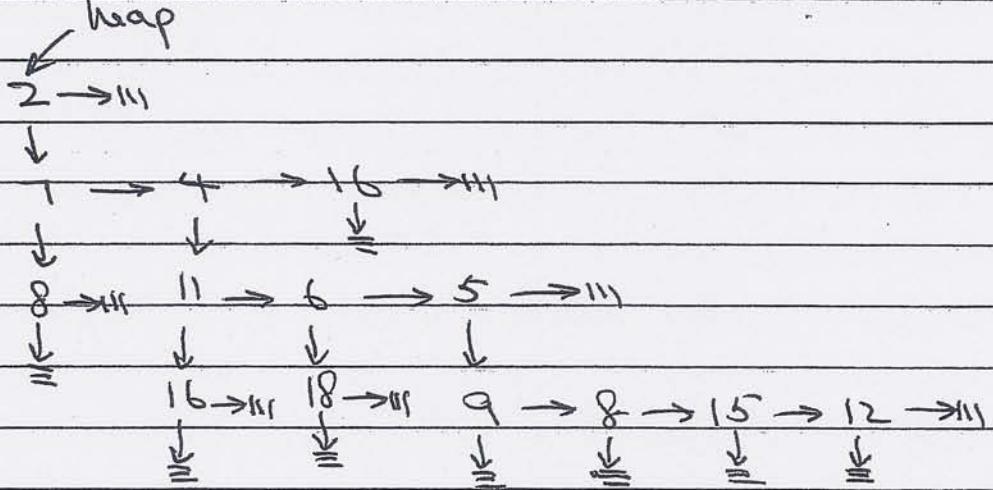
Two Pass method for merging:





Final pairing heap after deleteMin.

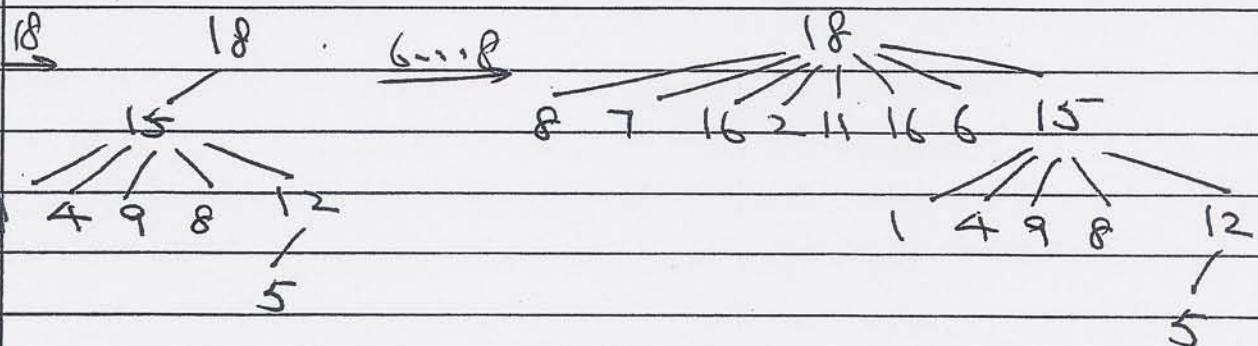
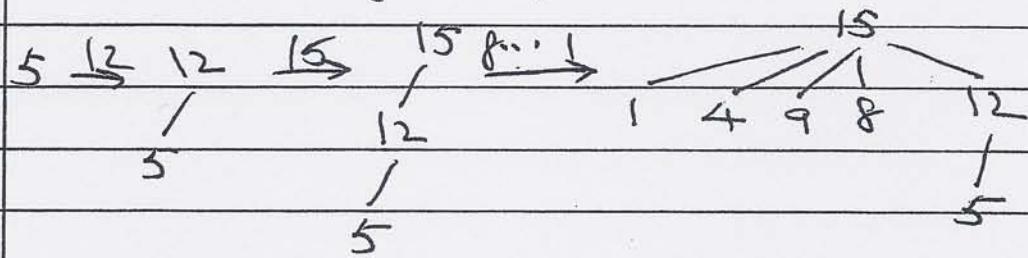
DS:



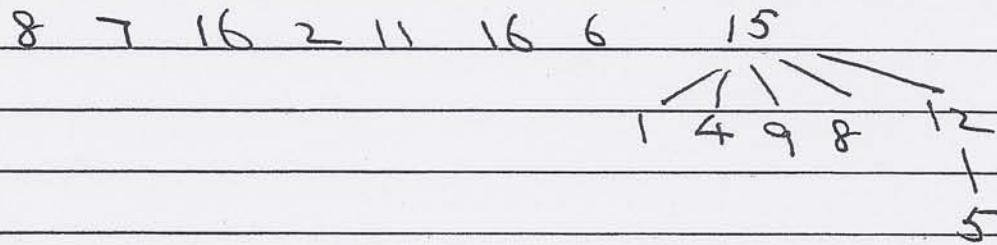
Either singly linked list, doubly linked list,
or circular doubly link list structures
is acceptable!

(28)

(j)

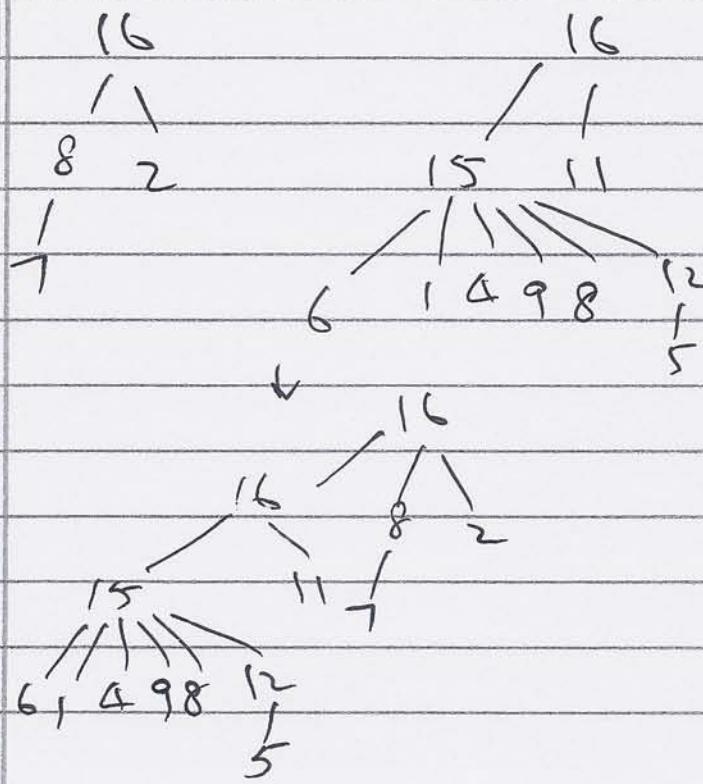
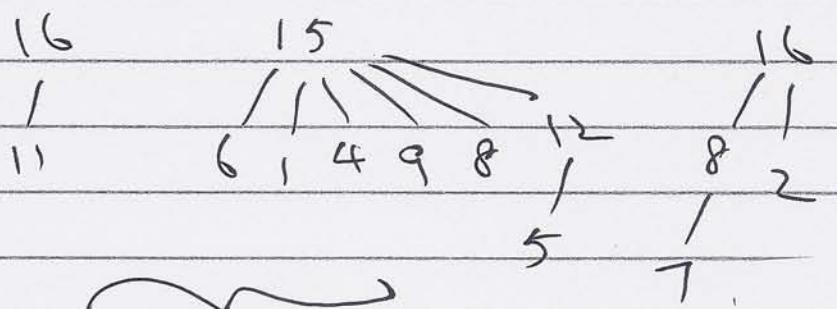
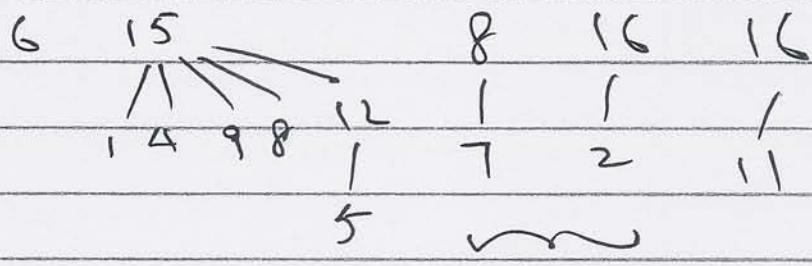
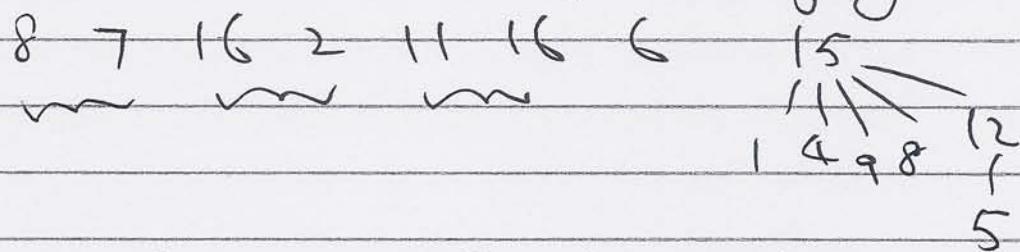
Max Pairing Heap:

Delete Max:



Multi-pass method for merging:

(29)



DS:

