Department of Computer Science
Data Structures (CSC211)
Final Exam (1st Semester 1428-29)
Date: 15/1/1429H (24-1-2008)

Time: 3 hours **Marks: 100**

Question 1 (20 marks) (إعارة كتب من المكتبة)
In this question you have to suggest suitable ADT for a simple library management system. A library requires a program for keeping track of its books. When a member of the library borrows a book, the librarian must be able to record that: (i) the book has been borrowed, (ii) the member who borrowed the book, (iii) the date the book was borrowed (تاريخ الاعارة), and (iv) the date the book is due back. When a book is returned the librarian must be able to record this information. At any time the librarian must be able to find out where a particular book is and what books a particular member has borrowed. Assume that each book has exactly one copy.

The ADT must not be tree based and must be described as follows:

(a) Draw a labeled graphical representation of the ADT showing its storage structure.
(b) Specify the following operations and for each operation specify which attributes of the ADT get updated by the operation:
   (i) Book_Borrowed() – librarian issuing a book to a member.
   (ii) Book_Returned() – a member returning a book to the library.
   (iii) Book_Location() – librarian checking the location of a book.
   (iv) Member_HasWhat() – librarian checking the books a member has borrowed.

Question 2. (10 marks)

(a) What is the time complexity (as step count) of the following code fragment? Give the count for each statement and the total count. Note that $\sum_{i=1}^{n} i = \frac{n(n+1)}{2}$.

```
for (int i = 0; i < n - 1; i++)
   for(int j = i + 1; j < n; j++) {
      tmp = a[i, j];
      a[i, j] = a[j, i];
      a[j, i] = tmp;
   }
```

(b) Find the Big-O for the following: $(2n^3 + n - 1)\log(2n^2 + 3)$

Question 3. (25 marks)

(a) Enqueue the following elements into a priority queue implemented as a heap and show the resulting priority queue in array format after each enqueue operation. Lower value means higher priority. The elements are: {13, 8, 5, 10, 4}.
(b) From the priority queue obtained in (a) serve (dequeue) four elements and show the resulting priority queue in array format after each serve operation.

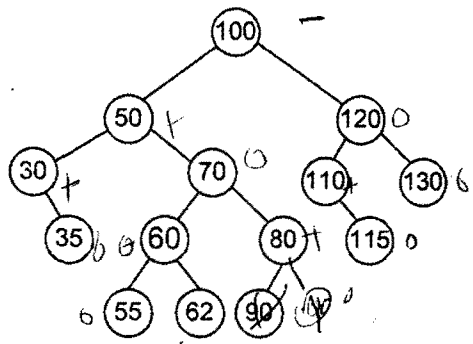
(c) Write HeapSort algorithm in pseudo-code. Use it to sort the following data in increasing order: {1, 11, 3, 7, 16}. Show the heap after each iteration.

(d) In a complete binary tree of height $h=3$, the minimum and maximum number of nodes are respectively 4 and 7. In a complete binary tree of height h: (i) Give a general formula to compute the minimum number of nodes (ii) Give a general formula to compute the maximum number of nodes.

Question 4. (15 marks)

(a) In the AVL tree shown on the right, insert key 61 and show the resulting tree.

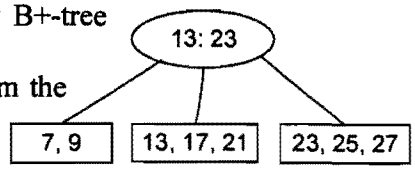
(b) Delete the keys 55, 62 and 115 from the AVL tree shown on the right and show the resulting tree.

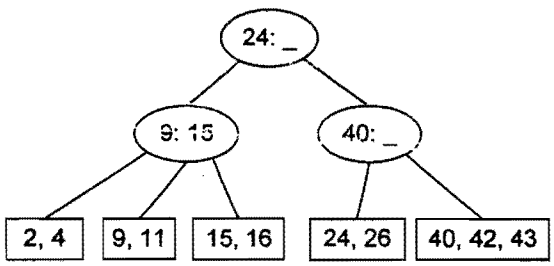


Question 5. (20 marks)

(a) Shown on the right is a B+-tree of order 3. Perform the insert operation for keys, 11, 28, 15, and 2. Show the resulting B+-tree after each insert.

(b) Shown below is a B+-tree of order 3. Perform the delete operation for keys, 26, 24, and 9. Show the resulting B+-tree after each delete.

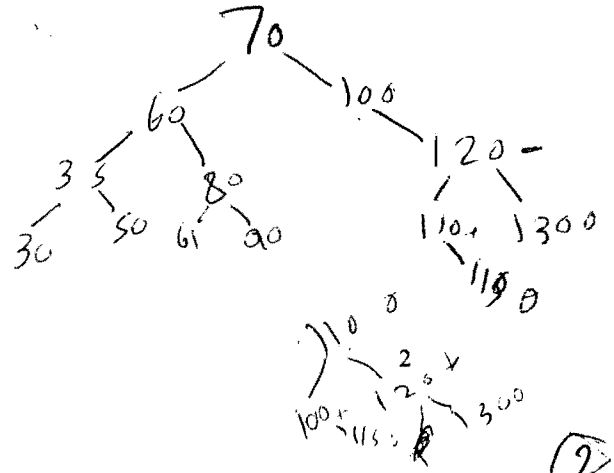




Question 6. (10 marks)

Add a recursive method to the LinkList class specified as follows:

Procedure numEvenElements (LinkList L)
Requires: List L is not empty. **Input:** List L.
Results: counts the number of nodes with even integer data. **Output:** number of nodes with even integer data.



Questions about heap from finals:

Q3: (a)
page 1

13, 8, 5, 10, 4

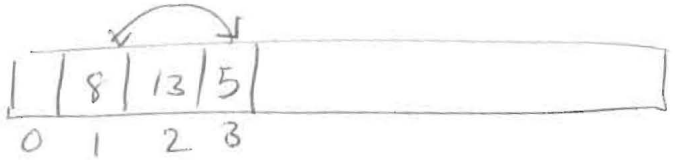
(13)

i = 1

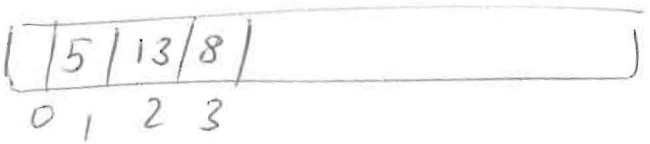
(8)

i = 2
parent = 1

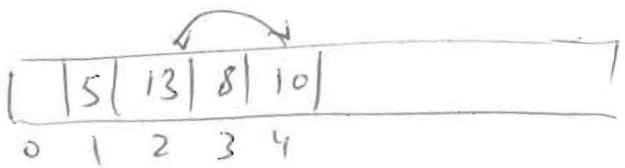
(5)



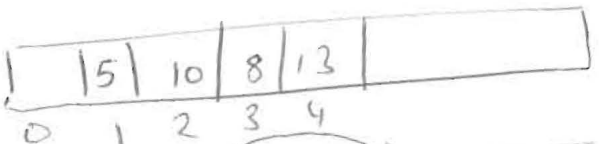
i = 3
parent = 1



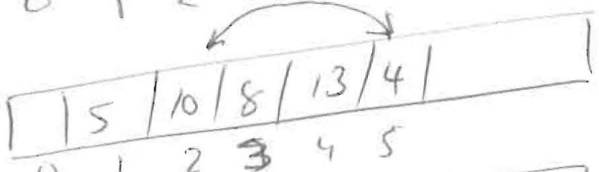
(10)



i = 4
parent = 2

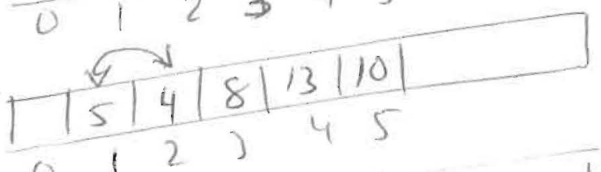


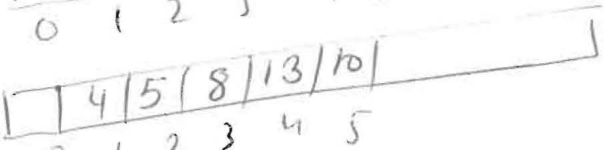
(4)



j = 5
parent = 2

i = 2
parent = 1

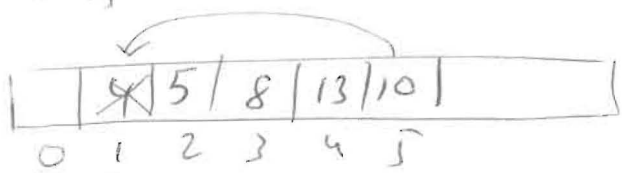


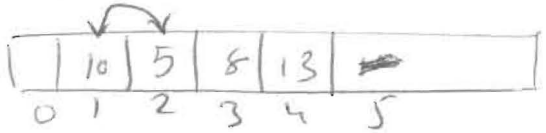


Q3, b) 1

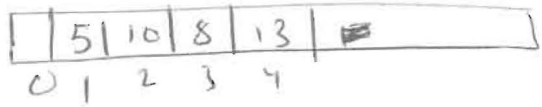
dequeue 4 elements:

first element



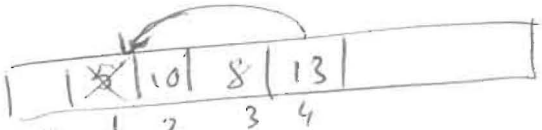


i = 1
children = 2, 3

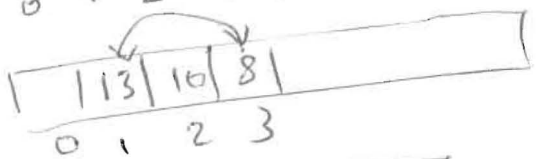


i = 2
children = 4, ~~8~~

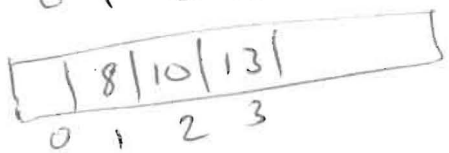
Second element



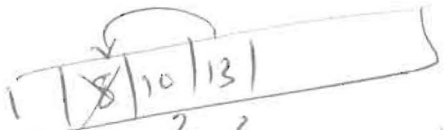
i = 1
children 2, 3



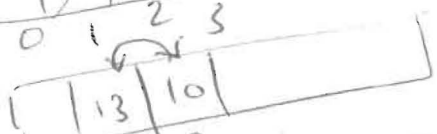
i = 3
children = ~~6~~, ~~7~~



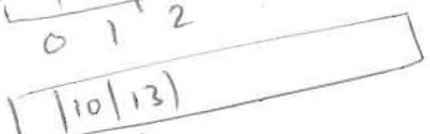
Third element



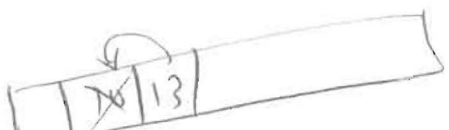
i = 1
children = 2, ~~3~~



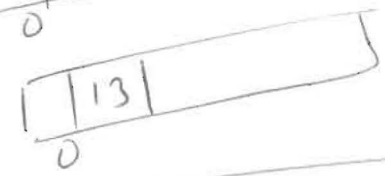
i = 1
children = 2, ~~3~~



fourth element



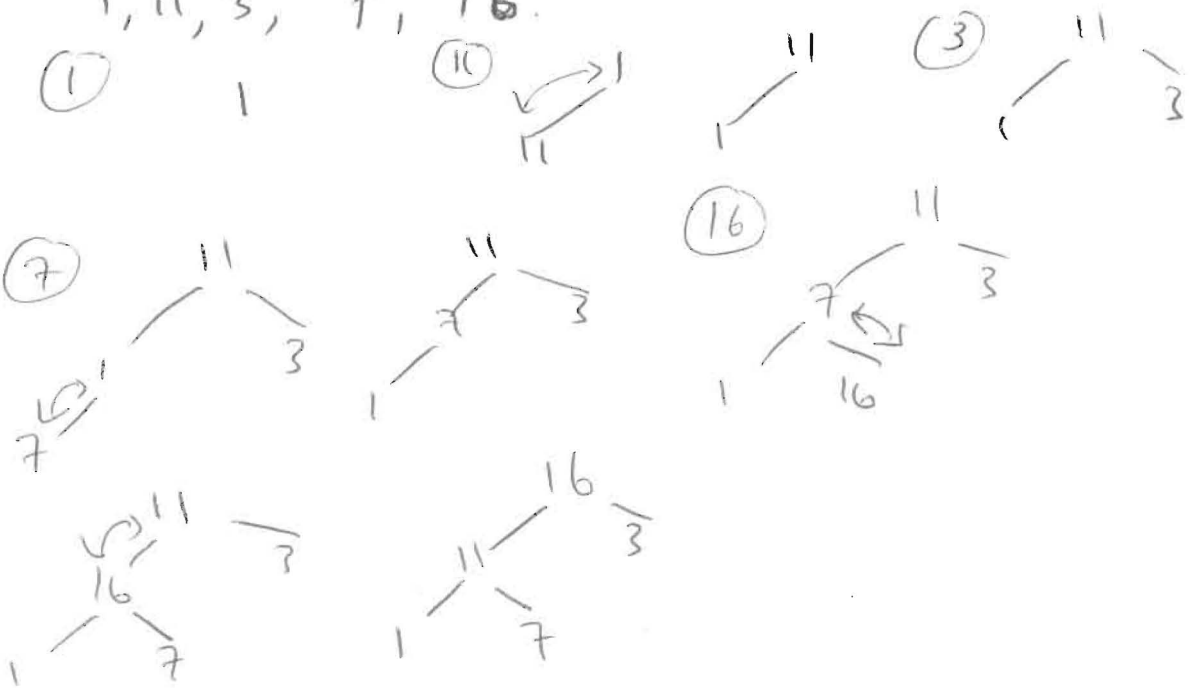
done



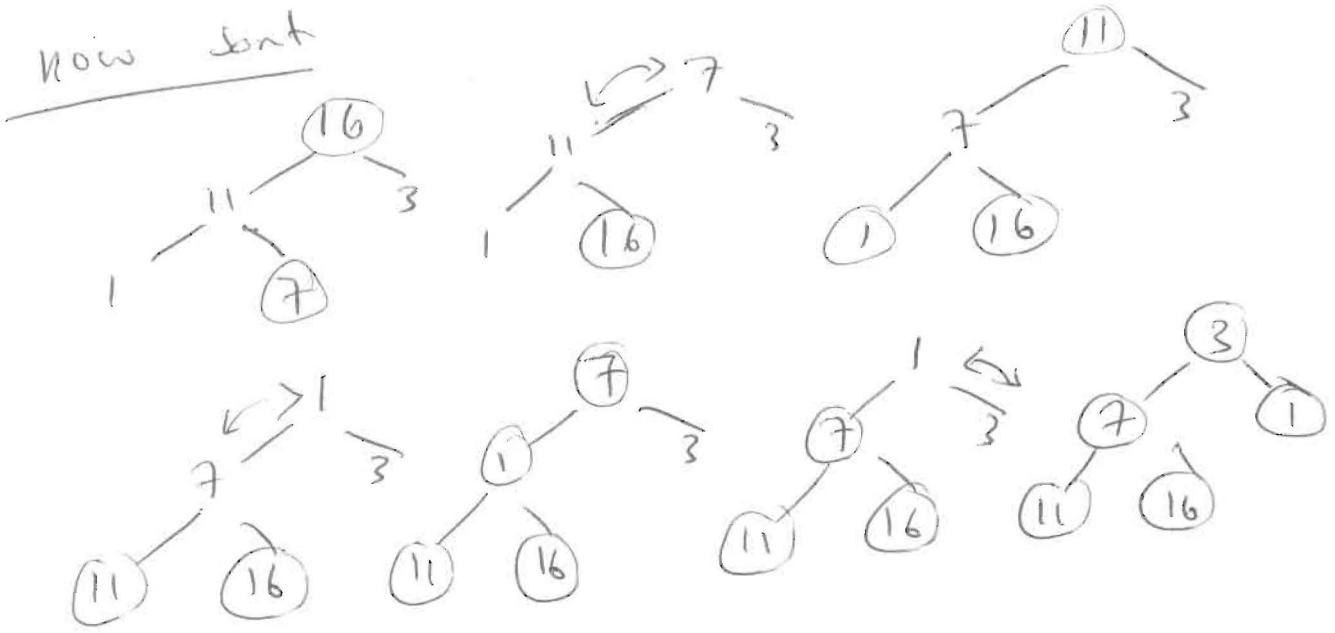
Q3-C)
page 1

```
while (size > 1)
{
    swap(heap, 1, size)
    size = size - 1;
    siftdown(heap, 1)
}
```

1, 11, 3, 7, 16.



now sort



8A

Q3-d: binary tree

page 1

$h = 3$

$max = 2^h - 1 = 2^3 - 1 = 8 - 1 = 7$

$min = h = 3.$

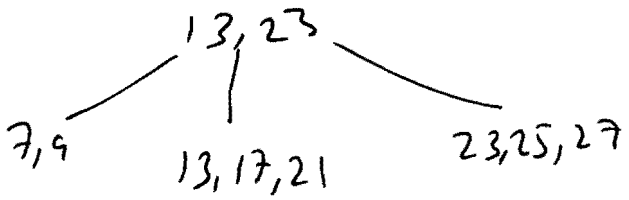
in Complete binary tree

(ii) $max = 2^h - 1 = 2^3 - 1 = 8 - 1 = 7$

(i) $min = 2^{(h-1)} = 2^2 = 4$.

Q5-a
page 2

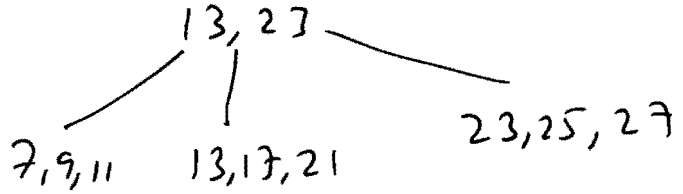
original tree with order 3



11, 28, 15, 2

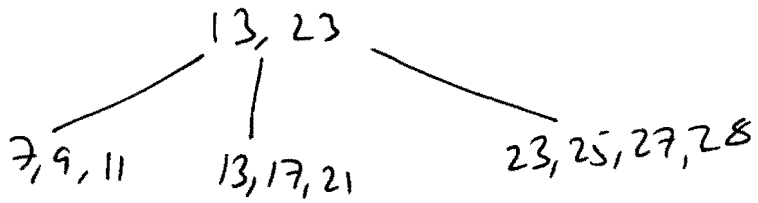
---

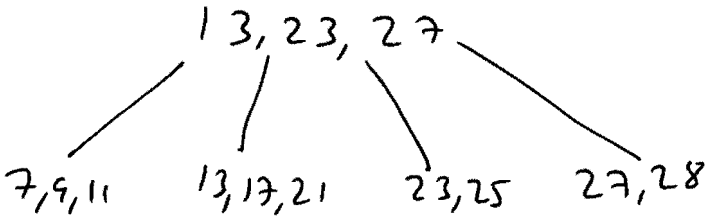
insent 11

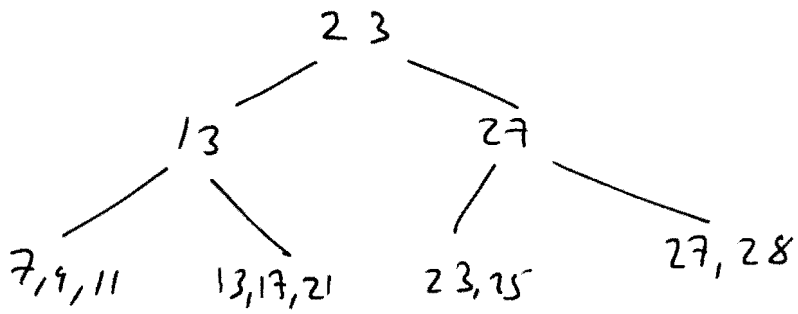


---

insent 28

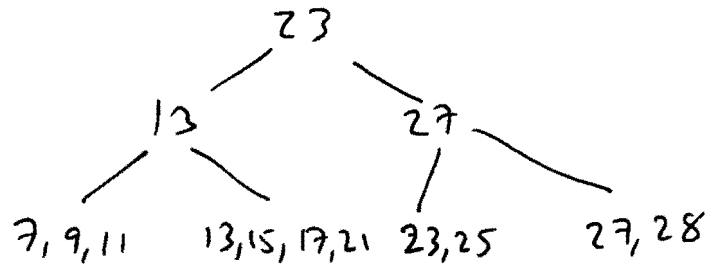


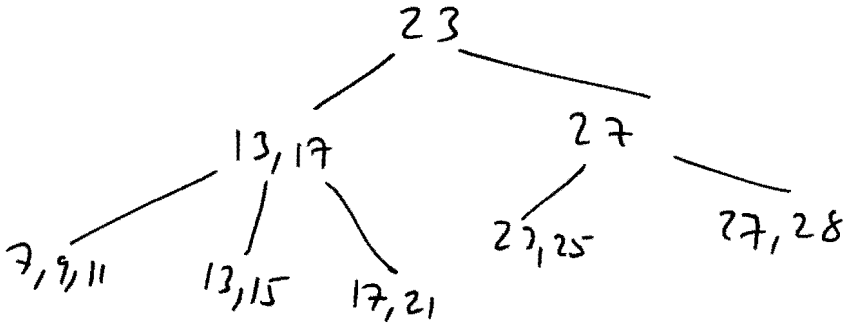




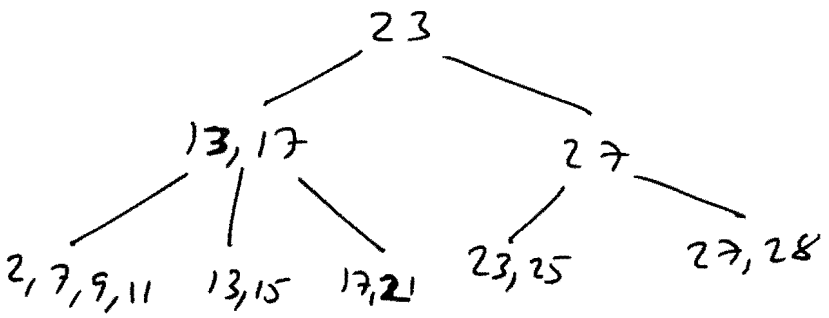
---

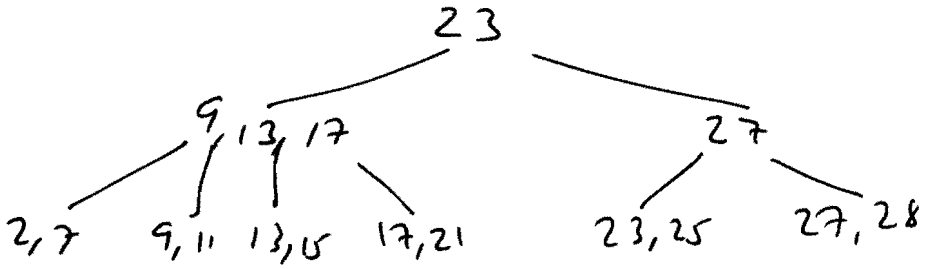
insent 15

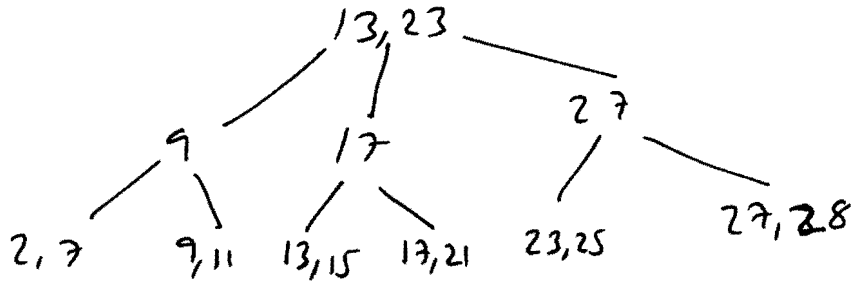



23
13,17
27
7,9,11
13,15
17,21
23,25
27,28

insent 2


23
13,17
27
2,7,9,11
13,15
17,21
23,25
27,28


23
9
13,17
27
2,7
9,11
13,15
17,21
23,25
27,28

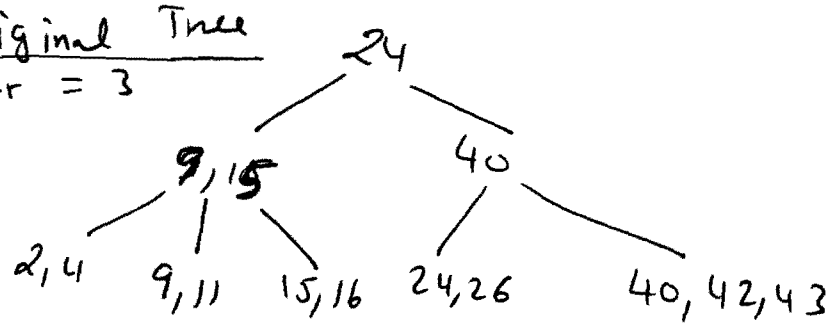

13,23
9
17
27
2,7
9,11
13,15
17,21
23,25
27,28

Q5-b

page 2

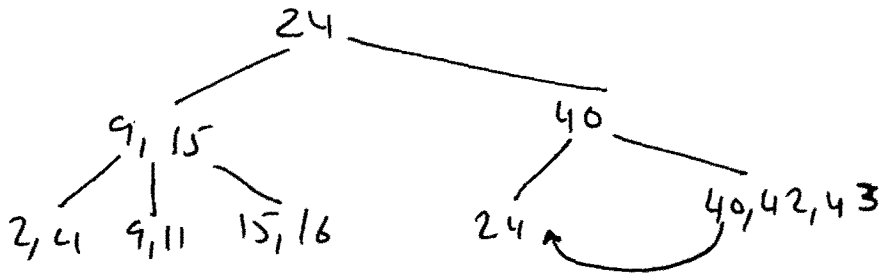
Original Tree

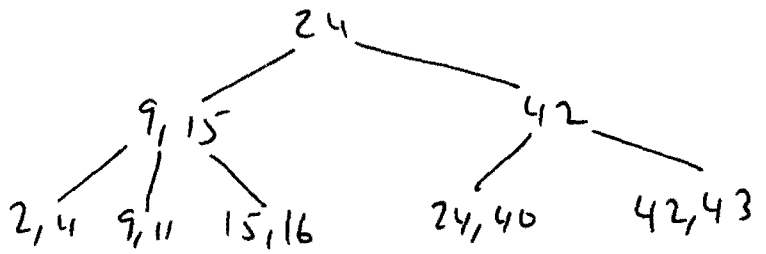
order = 3



delete : 26, 24, 9

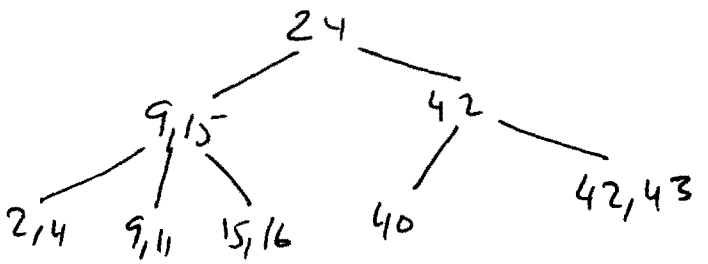
delete 26

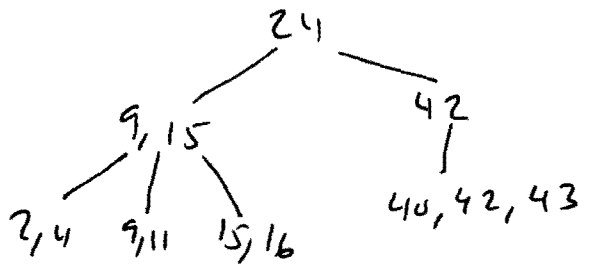


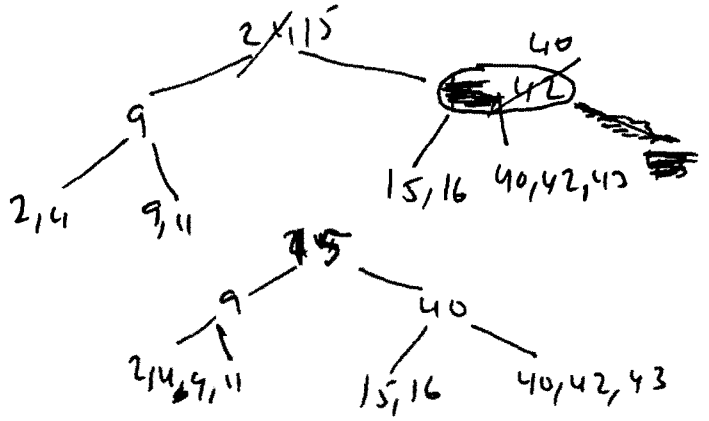


---

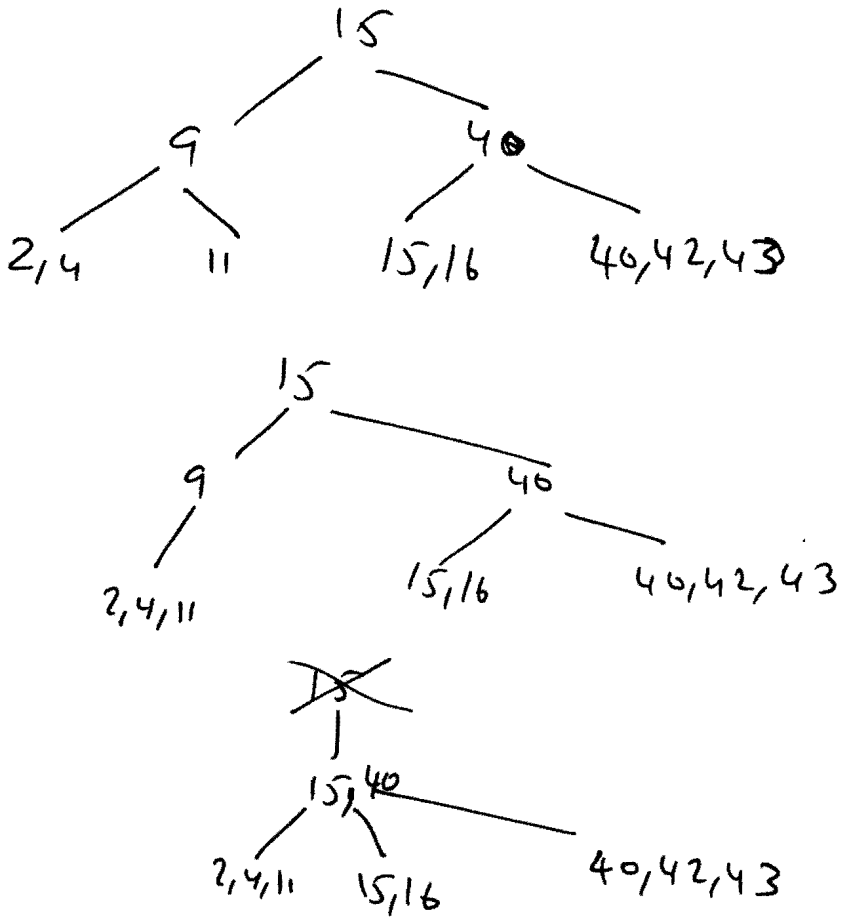
delete 24







delete 9


15
9
40
2,4
11
15,16
40,42,43
15
9
40
2,4,11
15,16
40,42,43
15
15,40
2,4,11
15,16
40,42,43