Department of Computer Science
King Saud University

Course title: CSC- 212 (Data Structures)
Semester: Fall 2008
Instructor: Dr. Muhammad Hussain, Dr. Ali El-Zaart, Dr. Inayatullah

Final Exam
SAFAR 10, 1430 (Feb. 5, 2009)

Time: 3 Hours **Marks: 100**

Question 1 (20)

(a) What is the time complexity of insert and remove operation in a doubly-linked list? O(1) O(1)

(b) What is the structure of a binary tree?

(c) What is $\log_2 128$ equal to? 7

(d) Why is the 'domain' of an ADT finite or bounded?

(e) Consider the insert operation in BSTs. Where will the 'current' end up after the operation, if the key we are trying to insert is already present in the BST?

(f) Can a heap have two nodes with the same priority? no

(g) Suppose a non-leaf node in a B+-tree has 100 key values stored in it. What is the number of children this node must be having? ~~[illegible] = 50~~ 100 +1 = 99

(h) Complete the following sentence. "A key stored in the non-leaf node of a B+-tree is the smallest key in the Root......."

(i) What is the advantage of implementing a priority queue as a heap?

Question 2. (5+5+5)

Consider the following situations and choose the most suitable ADT that can be used from the choices given. Give the reason in favour of your chosen ADT.

(a) You have to store information about books in a library (of 500,000 books). [[✓ AVL tree], B+-tree, List]

(b) You have to maintain information about the citizens (about 100,000,000) in a country. The frequent operations are to find the oldest citizen/youngest citizen, to access the information about any citizen, to add a new citizen and update the information of any citizen. Each citizen is allocated a unique national ID. [AVL-tree, [✓ B+-tree], BST, stack]

(c) You have to maintain information about people in a large family or a tribe of about 200 people. In particular, given a key of a person you should be able to find information about the person's father and mother quickly. Draw a graphical representation of the ADT you choose, labeling it clearly. [BST, AVL-tree, [✓ Binary Tree], B+-tree]

Question 3. (8+4+3)

(a) Find the time function T(*n*) of the following algorithm and express it in Big Oh notation.

O(n log n)

```
function  Algo(n) {
        for(int i = 0;  i < n;  i++){
                for(int j = i;  j< n; j++)
                        S1;
                for(int k = 0;  k< i; ~~j~~k++){
                        S2;
                        S3;
                        S4;
                }
        }
}
```

Note: For each for loop, consider only comparison statement. e.g. to count the number of iterations of the first for loop, consider only "i < n".

(b) Find the big-*O* of $n^2 + 2^n + 16\log n$. What are the values of *C* and n_0 ?

$n^2 + 2n^2 + 16n^2 = 19n^2$ $n \geq 1$ $C = 19$ $n_0 \geq 1$

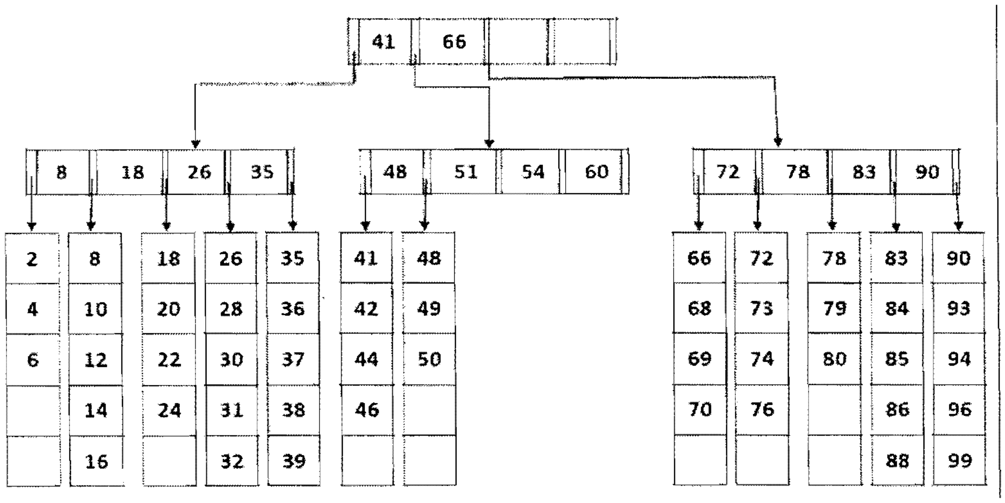
(c) Time complexity of the algorithm A is O(*n*log*n*) and that of the algorithm B is O(n^2). Which algorithm is more efficient?   O(nlogn)

Question 4. (5+6+6)

(a) Is the following tree B⁺- tree of order 5 (M = 5)? If not, give the reason why not.

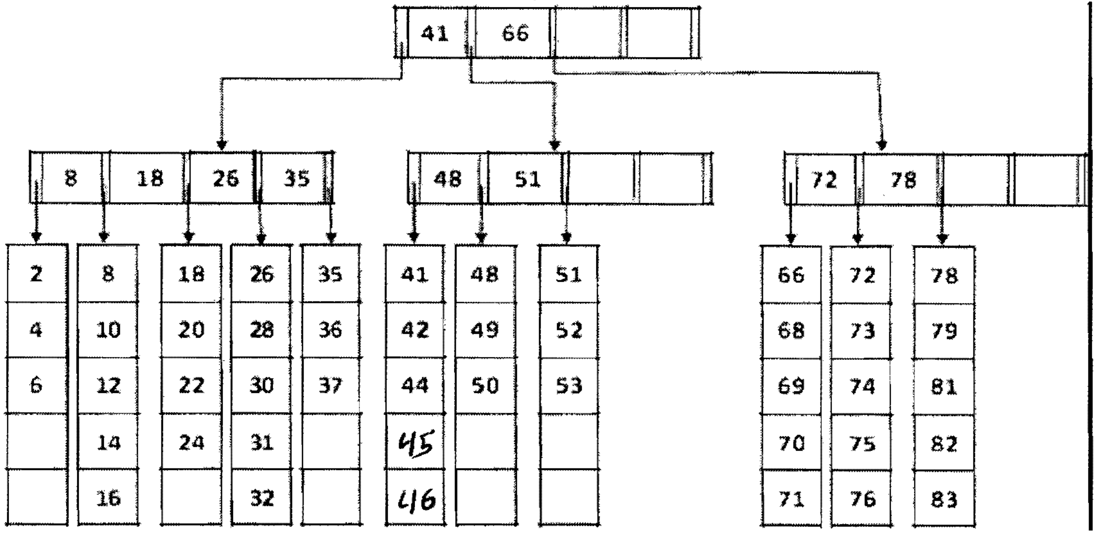
m/2

3 5/2

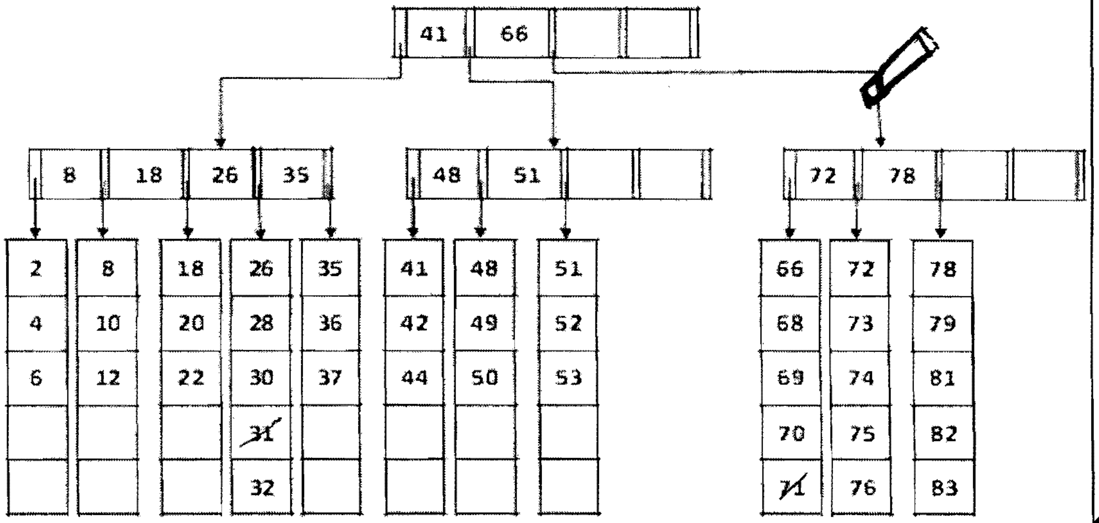


becose the min is 5/2 = 3 and the ~~B~~ Node 48 51 54 60 they have only 2 Clidren.

(b) In the following B^+- tree of order 5 (M = 5), insert 45, 46, 77



(c) From the following B^+-tree of order 5 (M = 5) remove 12, 31, 71



Question 5. (8+8)

(a) Consider the sorted array 10, 12, 20, 30, 50, 70. Use heap to sort this array into decreasing order. Show the steps in diagrams.

(b) Write a static method that removes all elements from a priority queue whose key is less than *k*.

Question 6. (8+5+4)

Given the values: 10, 20, 30, 40, 50, 60, 70, 80, 90, and 100, 65.

(a) Insert these values in an empty AVL tree in the given order and show the final AVL tree in a diagram

(b) Tell the number of single and double rotations which you performed during the insert operations of part (a).

(c) Delete 50 form the resulting AVL tree of part (a).

Question 1 (20)

(a) What is the time complexity of insert and remove operation in a doubly-linked list?
O(1) for both insert and remove.

(b) What is the structure of a binary tree?
Nested or a hierarchical.

(c) What is $\log_2 128$ equal to?
7

(d) Why is the 'domain' of an ADT finite or bounded?
Because of physical memory.

(e) Consider the insert operation in BSTs. Where will the 'current' end up after the operation, if the key we are trying to insert is already present in the BST?

It will not change.

(f) Can a heap have two nodes with the same priority?
Yes.

(g) Suppose a non-leaf node in a B+-tree has 100 key values stored in it. What is the number of Children this node must be having?

The non-leaf node (Index) should have keys + 1 child, so the answer is 101 children.

(h) Complete the following sentence. "A key stored in the non-leaf node of a B+-tree is the smallest key in the second split non-leaf node.

(i) What is the advantage of implementing a priority queue as a heap?

Both insert and delete has O(log(n)).

Consider the following situations and choose the most suitable ADT that can be used from the choices given. Give the reason in favor of your chosen ADT.

(a) You have to store information about books in a library (of 500,000 books). [AVL tree, B+-tree, List].

(b) You have to maintain information about the citizens (about 100,000,000) in a country. The frequent operations are to find the oldest citizen/youngest citizen, to access the information about any citizen, to add a new citizen and update the information of any citizen. Each citizen is allocated a unique national ID. [AVL tree, B+-tree, BST, stack].

(c) You have to maintain information about people in a large family or a tribe of about 200 people. In particular, given a key of a person you should be able to find information about the person's father and mother quickly. Draw a graphical representation of the ADT you choose, labeling it clearly. [BST, AVL-tree, Binary Tree, B+-tree].

Q3
a

```
                              ↙ n+1
for (int i =0; i<n; i++){
sched 1  ←——  for(int j=i; j<n; j++)
                   S1;

sched 2  ←——  for (int K=0 ; K<i; K++){
                   S2;
                   S3;
                   S4;
              }
}
```

sched 1

suppose n = 5

i	j	j<n	S1
0	0	6	5
1	1	5	4
2	2	4	3
3	3	3	2
4	4	2	1
5			

$\Rightarrow$ $\sum_{i=1}^{n} i = 1+2+3+\cdots+(n-1)+n = \frac{n(n+1)}{2}$

$j<n = \underbrace{2+3+\cdots+(n-1)+(n)+(n+1)}_{\frac{n(n+1)}{2}+n}$

$S1 = 1+2+3+\cdots+(n-1)+n = \frac{n(n+1)}{2}$

sched 2

suppose n = 5

i	K	K<i	S1	S2	S3
0	0	1	0	0	0
1	0	2	1	1	1
2	0	3	2	2	2
3	0	4	3	3	3
4	0	5	4	4	4
5					

$\Rightarrow$ $\sum_{i=1}^{n} i = 1+2+3+\cdots+(n-1)+n =$

$K<i = 1+2+3+4+5+\cdots+(n-1)+n = \frac{n(n+1)}{2}$

$S1 = 0+1+2+\cdots(n-1) \approx \frac{n(n+1)}{2} - n$

S2 & S3 same as S1.

$T(n) = (n+1) + \left(\frac{n(n+1)}{2} + n\right) + \left(\frac{n(n+1)}{2}\right) + \left(\frac{n(n+1)}{2} + 3(\frac{n(n+1)}{2} -$

$= n+1 + \overset{3}{\cancel{6}}\left(\frac{n(n+1)}{\cancel{2}}\right) + n \cancel{+} 3n$

$= n+1 + 3(n^2+n) + n - 3n$

$= n+1 + 3n^2 \cancel{+3n} + n \cancel{-3n}$

$= \boxed{3n^2 + 2n + 1}$

$f(n) = 3n^2+2n+1 \leq 3n^2+2n^2+n^2$ $\quad n_0 = 1$

$\leq 6n^2$ $\quad n_0 = 1$

$O(n^2)$ $\quad c = 6$ $\quad n_0 = 1$

Q 3/b $\quad n^2 + 2^n + 16 \log n$

$n^2 <= n^2 \Rightarrow O(n^2)$ $\quad c = 0$ $\quad n_0 \geq 0$

$2^n <= 2^n \Rightarrow O(2^n)$ $\quad c = 0$ $\quad n_0 \geq 0$

$16 \log n <= 16 \log n \rightarrow O(\log n)$ $\quad c = 16$ $\quad n_0 \geq 0.$

$O(n^2) + O(2^n) + O(16 \log n) = O(2^n)$

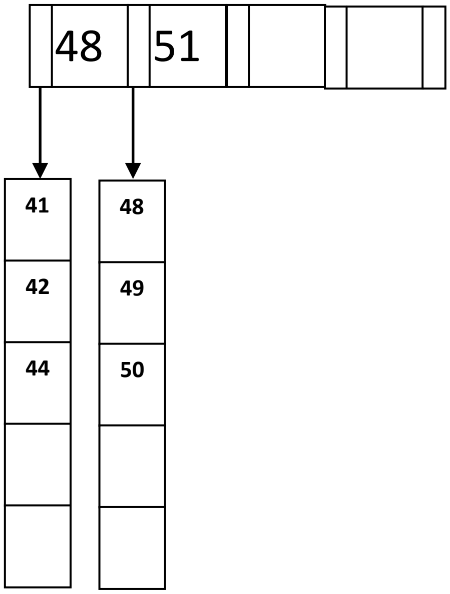
$c = 0$ $\quad n_0 \geq 0.$

Q 3/c

$O(n \log n)$ is more efficient.

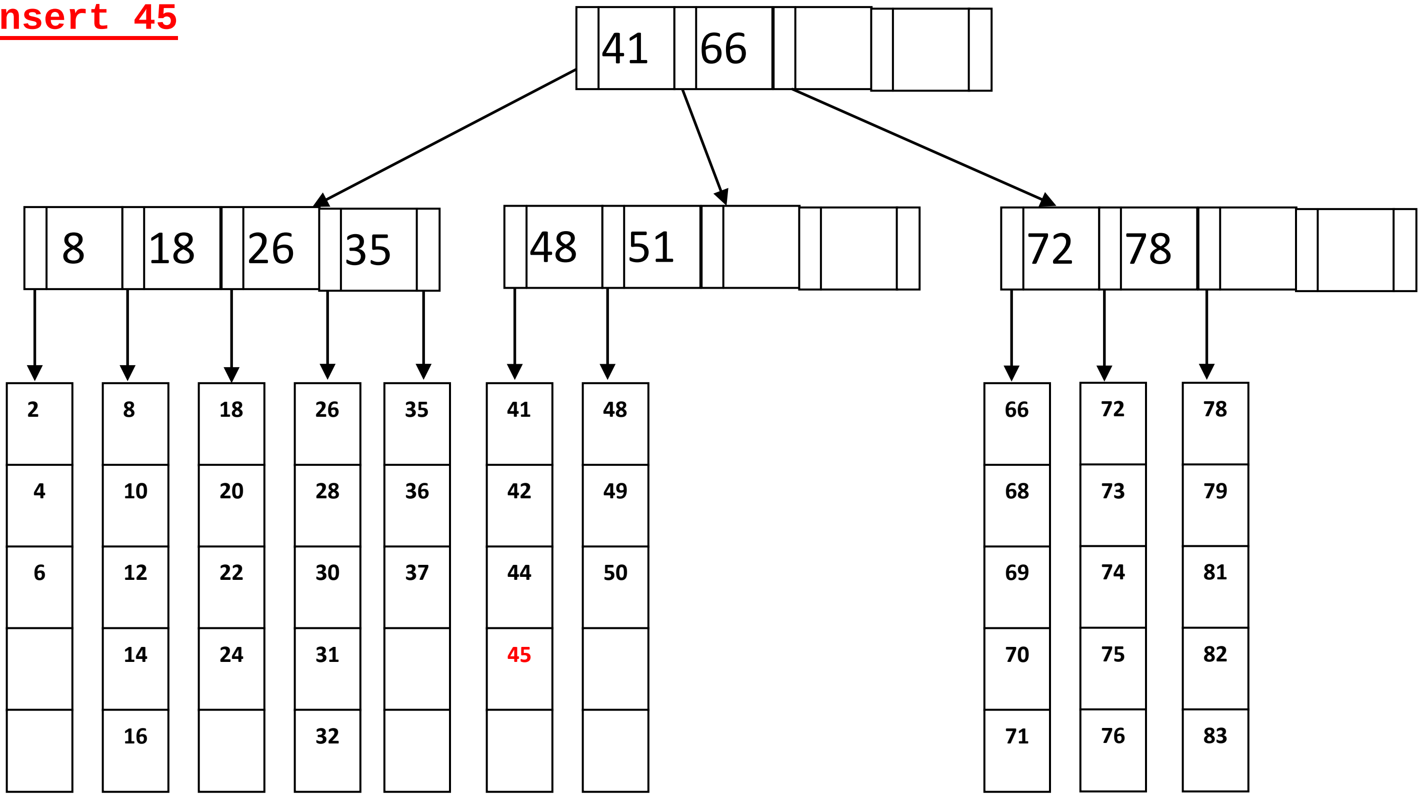
Q4)-a

The tree is not B+-tree because the Order is 5 which means the index must have minimum 3 children, but we have node with only 2 children.



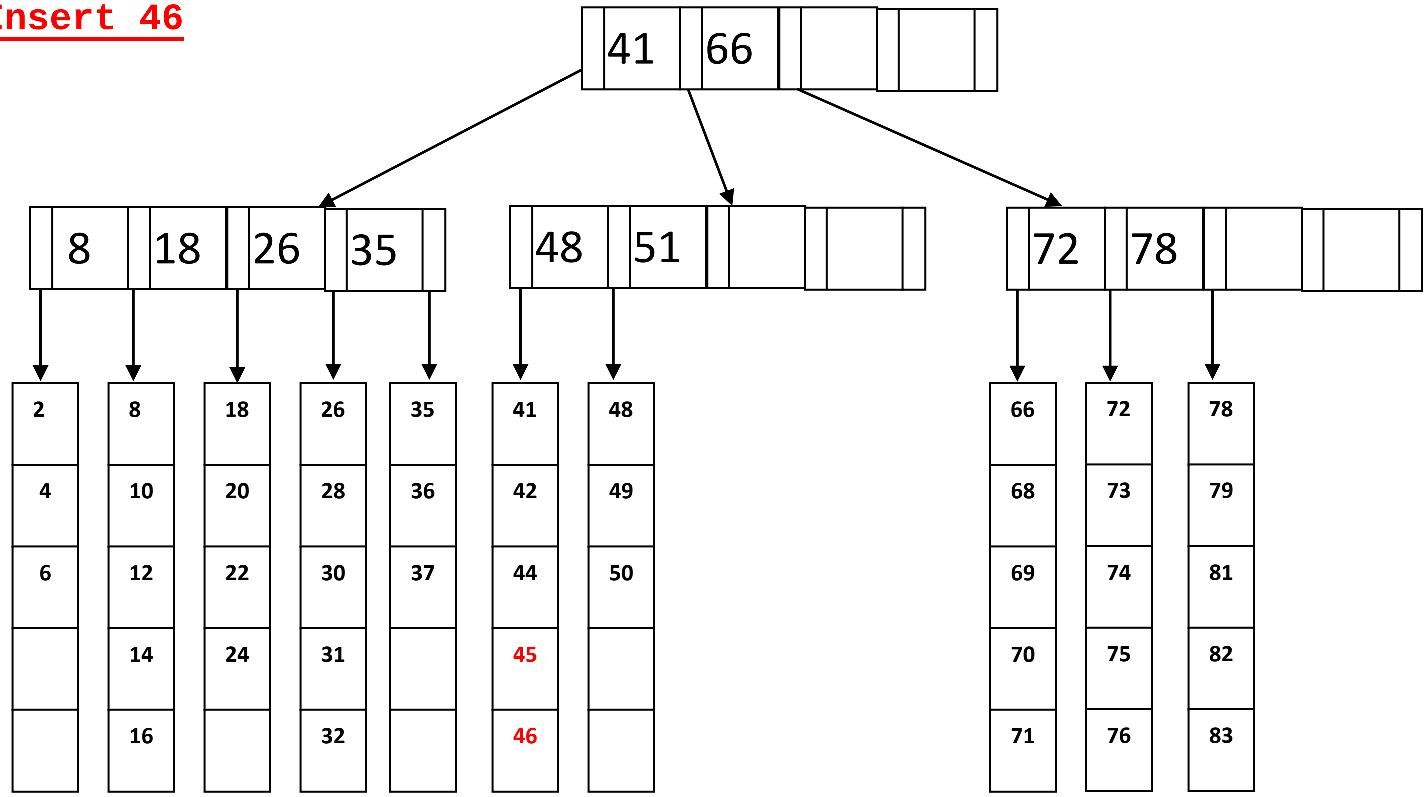
Q4)-b

Insert 45



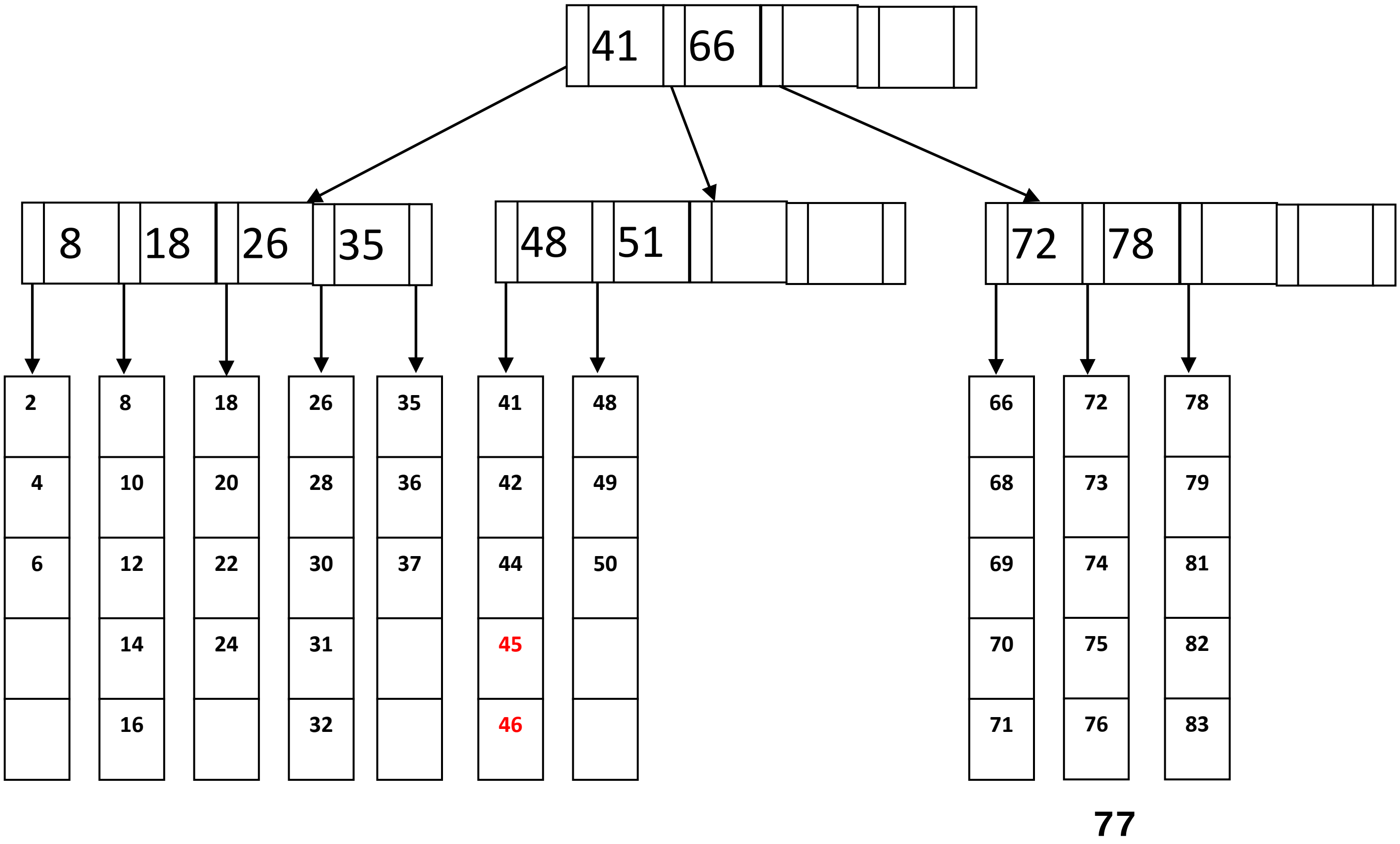
After inserting 45 in leaf node, we have 4 data which less than 5 (the order of the tree)

Insert 46

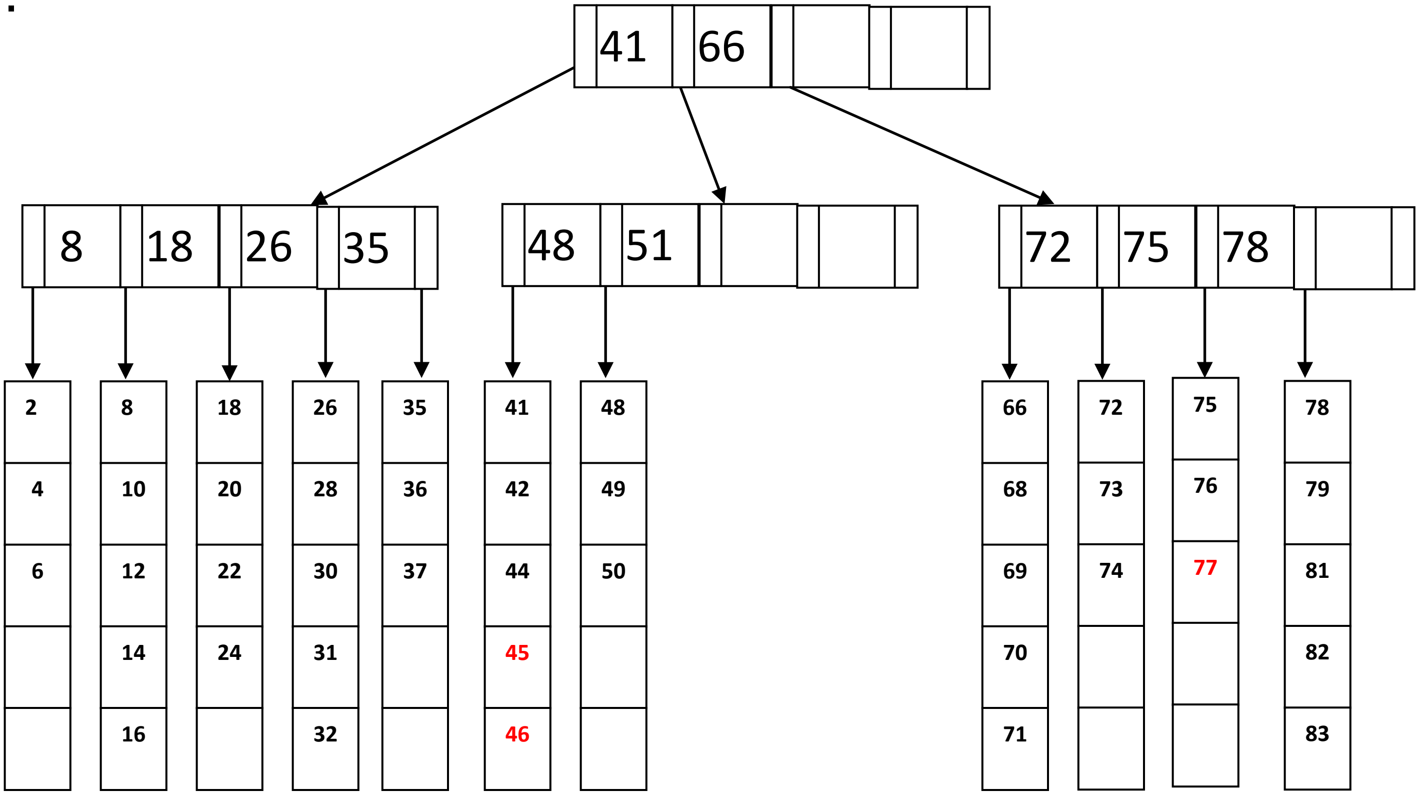


After inserting 46 in leaf node, we have 5 data which equals 5 (the order of the tree)

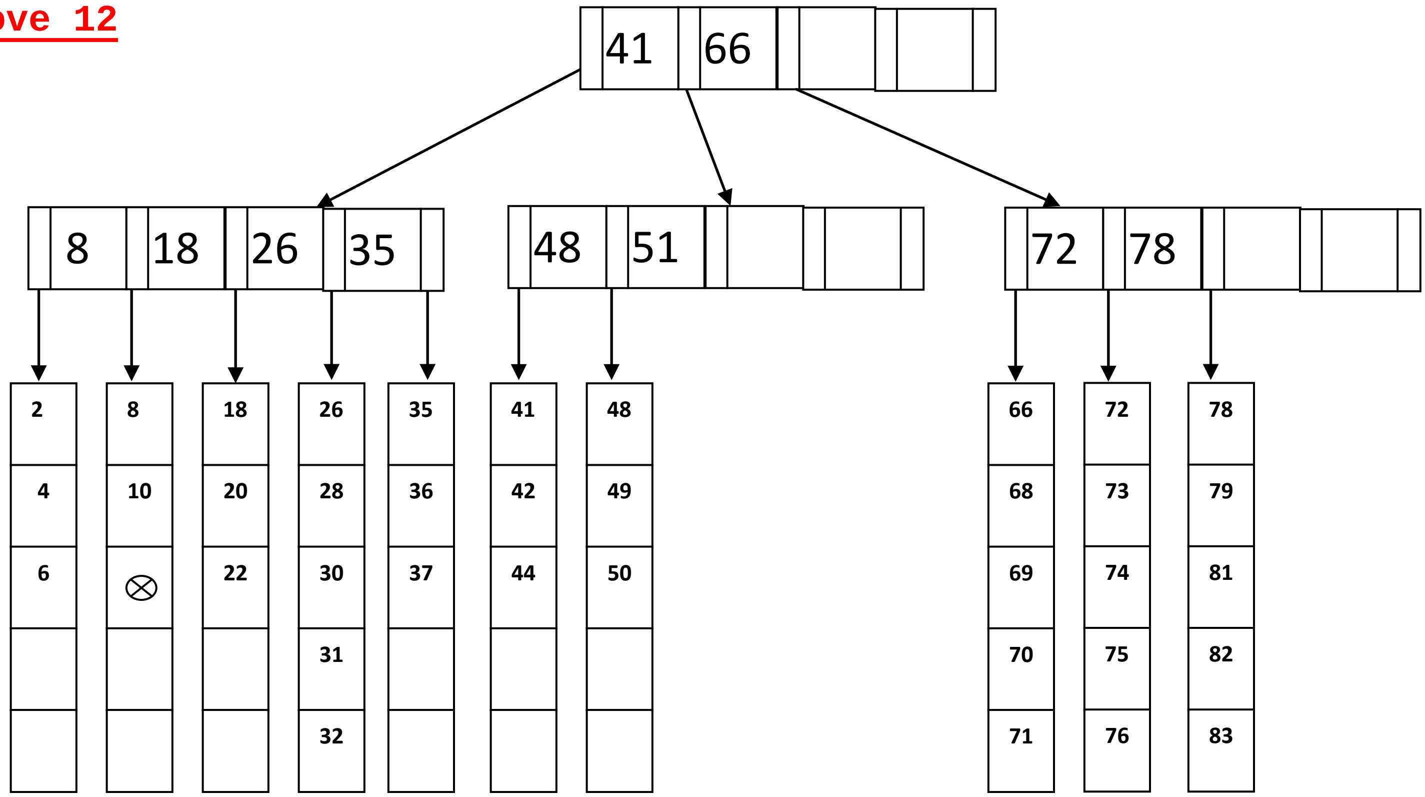
Insert 77



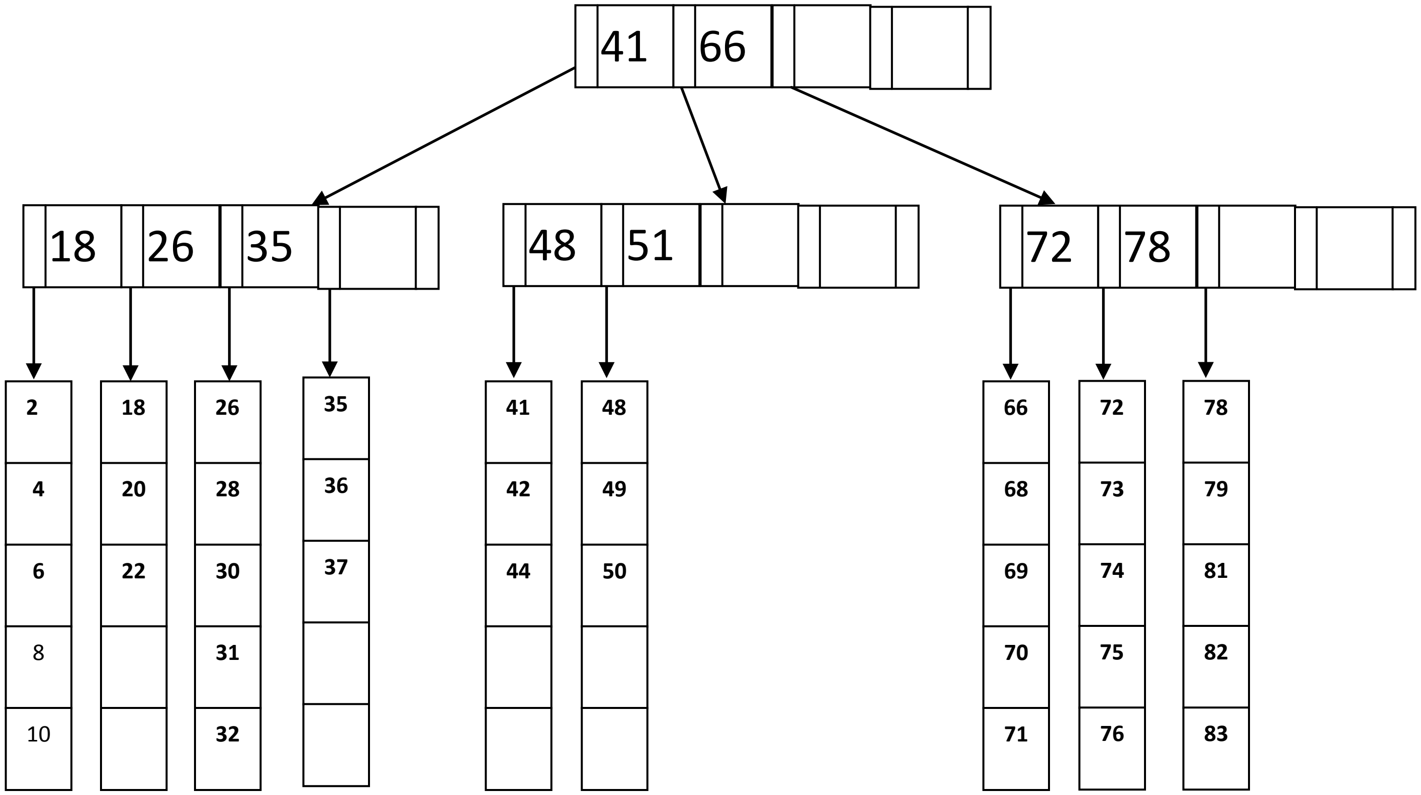
The node is over, it has 6 data and m = 5, we have to split the leaf node.



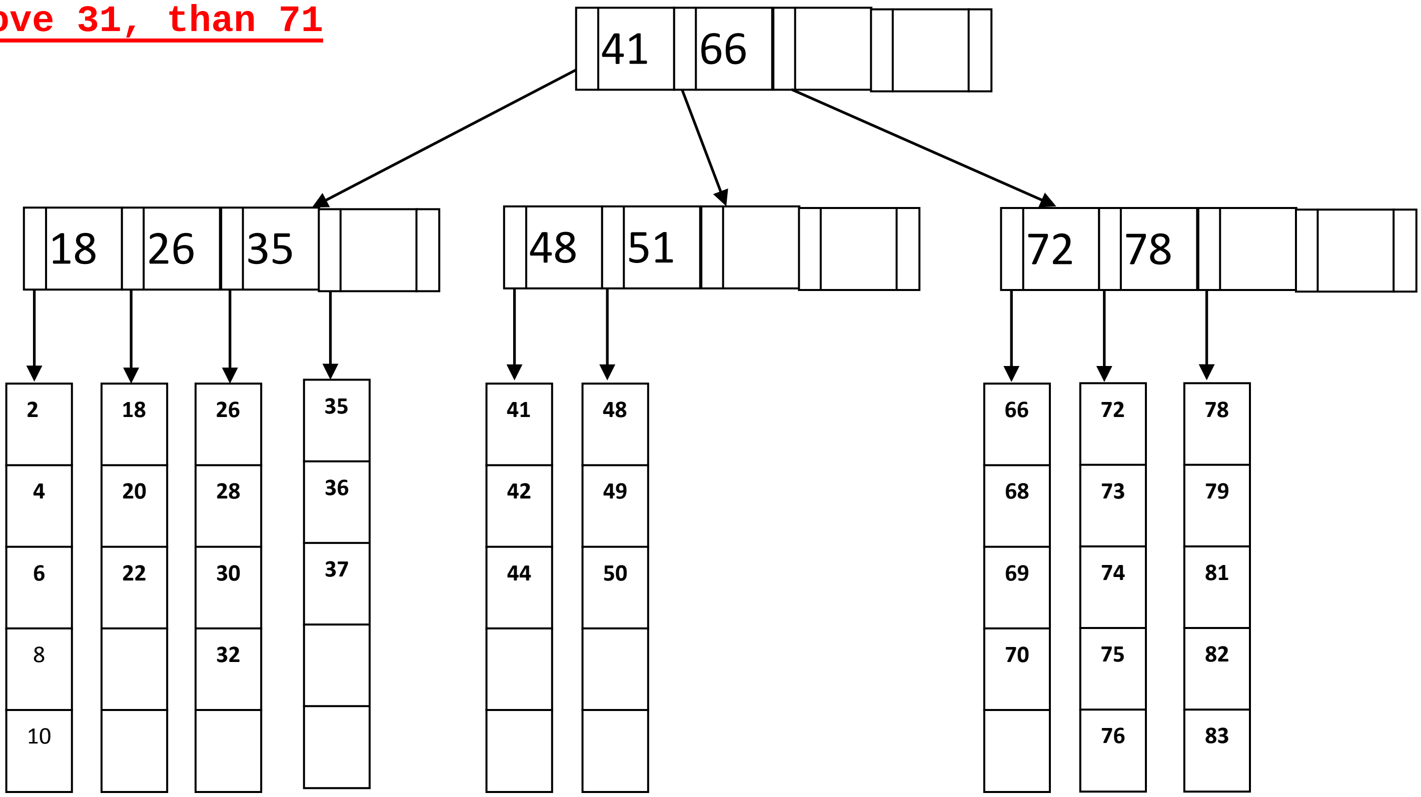
Remove 12



After deleting 12 we got 2 data in the leaf node, the order = 5 , minimum is 3, we will try to borrow from sibling, we cannot, than we do merge



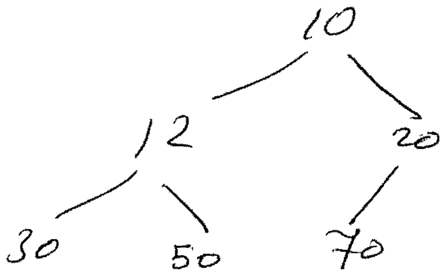
Remove 31, than 71



Q5 Array

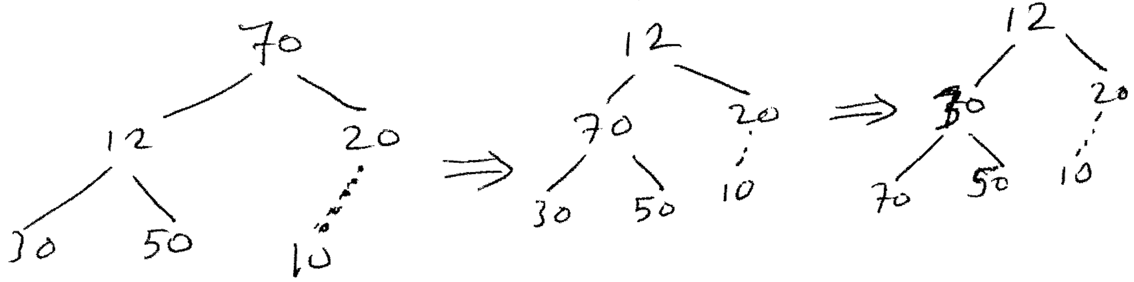
a

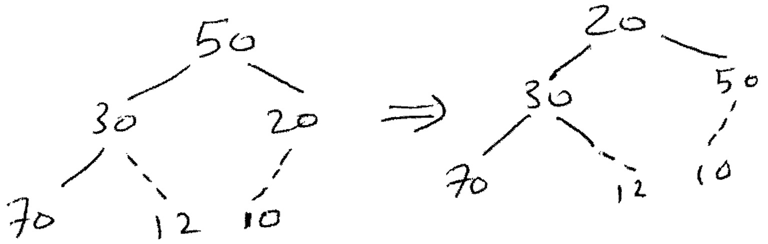
10	12	20	30	50	70

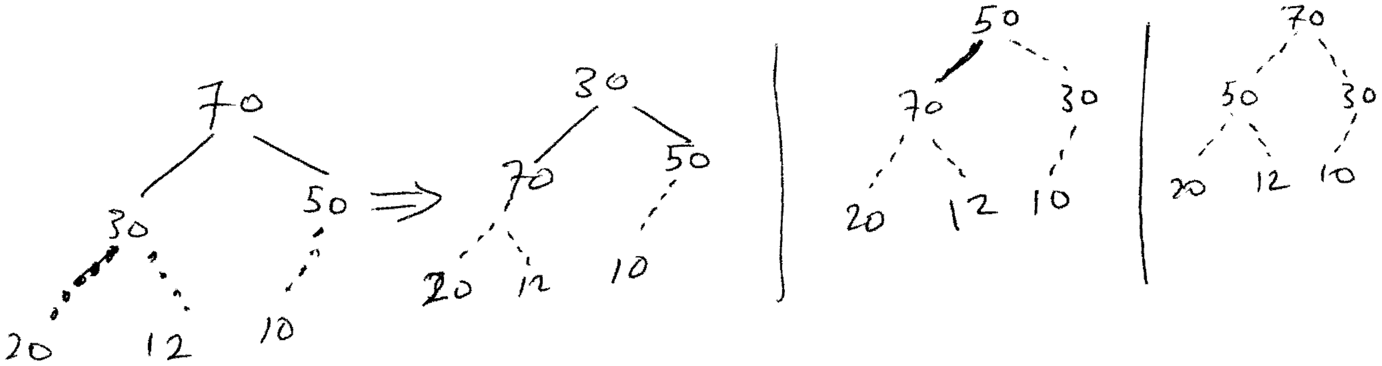


Sort Algorthim

```
while (size !=0)
{    swap(root, last);
     size--;
     siftdown(root);
}
```

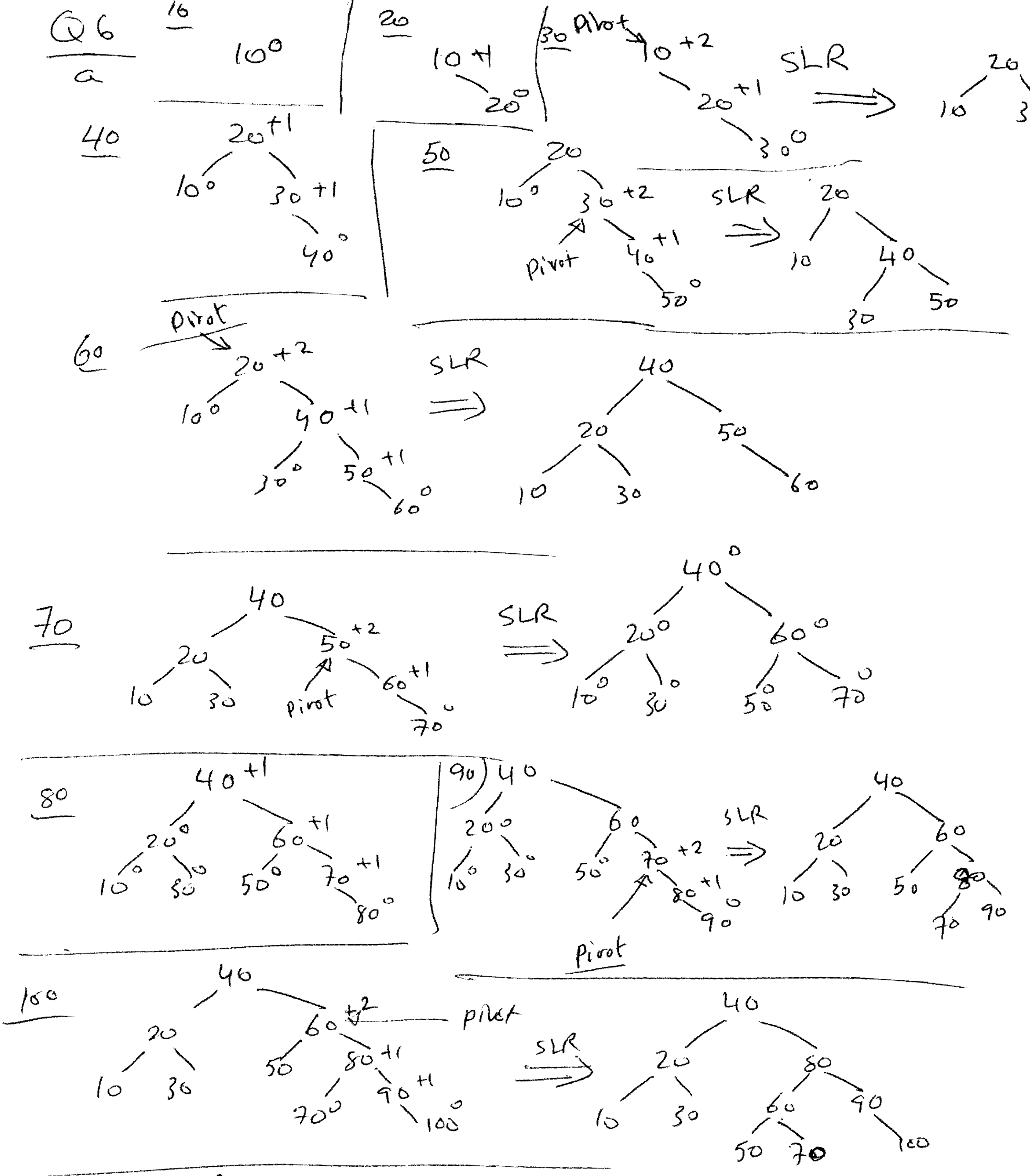


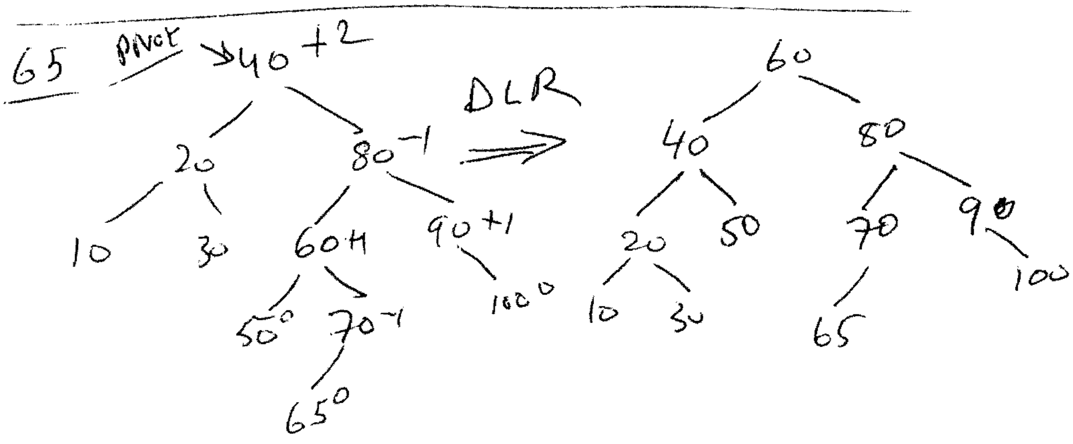




70	50	30	20	12	10

Q6
a





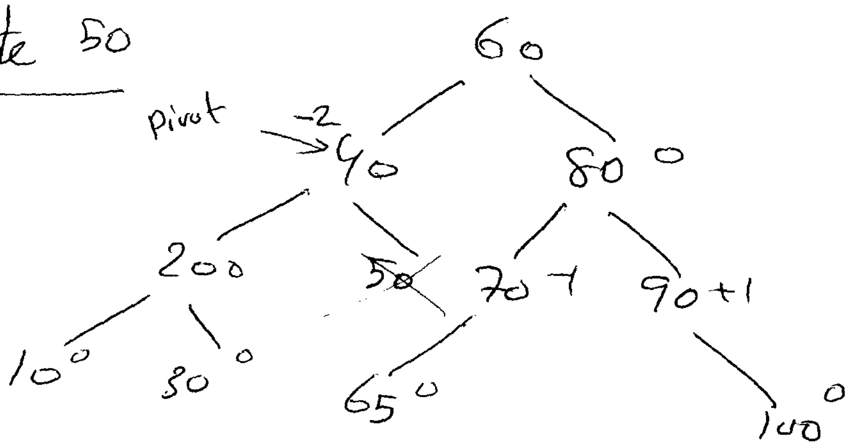
Q6/b

Single 6

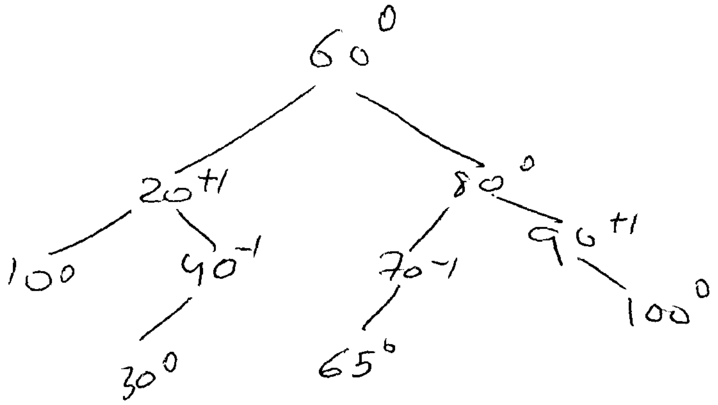
Doble 1

Q6 / C

delete 50



Left, Left



60 0
20 +1
80 0
10 0
40 -1
70 -1
90 +1
100 0
30 0
65 0