CLEVELAND STATE UNIVERSITY

CIS 606 – ANALYSIS OF ALGORITHMS

MID TERM

Name: Prudhvi Reddy Araga

Not using filename mid.pdf: -5

Login ID: praraga

39/100

Question 1:

a) $5^{n+5} = O(5^n)$

Answer: True

Let us assume that $f(n) = 5^{n+5}$ and $g(n) = 5^n$

$f(n) \in O(g(n))$ for two constant c and n_0 such that $0 \leq f(n) \leq c \, . \, g(n)$, for all $n \geq n_0$

use \cdot

➔ $0 \leq 5^{n+5} \leq c. 5^n$

As per the Math formula, $a^{m+n} = a^m . a^n$

➔ $5^n . 5^5 \leq c. 5^n$

-1

➔ $c >= 5^5$; for $n >= 1$.

➔ $c = 3125$.

Since there exists two positive constants $c = 3125$ and $n_0 = 1$ such that $0 \leq 5^{n+5} \leq 3125. 5^n$; for $n >= 1$. Therefore, $5^{n+5} = O(5^n)$.

c

n_0

b) In the algorithm SELECT which uses the median of medians, the input elements are divided into groups of 5. The algorithm can still work in linear time if they are divided into groups of 9.

Answer: True

at least

If the linear time is divided into groups of 9 then we will have n/9 groups with $\cancel{4}.(1/2).(n/9) = 2n/9$ elements.

at most

Remaining elements $= n - 2n/9 \cancel{=} 7n/9$

$T(n) = T(n/9) + T(7n/9) + n$

-6

As per the theorem $T(n) = T(\alpha n) + T(\beta n) + n$ where $\alpha + \beta < 1$

$$T(n) = O(n)$$

c) Finding a closest pair of points in 2 dimensions would be harder (in terms of time complexity) if the distance between 2 points (x1, y1) and (x2, y2) were defined as $|x1 - x2| + |y1 - y2|$.

Answer: False

As per the divide and conquer approach, we find the delta of each dimension and pick the minimum of both in order to calculate the closest pair.

The closest pair distance for P_L and P_R will be equal to min (Delta L, Delta R).

T(n) = 2 T(n/2) + $\theta(n)$ (2 points are given)

Let a = 2, b= 2 and f(n) = $\theta(n)$

-7

$= \theta(n^{\log_b a})$

$= n^{\log_2 2} = n$

If we use the approach of brute force technique, then the time complexity is $\theta(n^2)$.

As per the merge sorting algorithm, the time complexity is $\theta(n \log n)$ which is better than the brute force algorithm that uses $\theta(n^2)$.

Question 2:

Solve the following recurrence by making a change of variables. T(n) = 8T(√n) + 1

Given equation $T(n) = 8\, T(\sqrt{n}) + 1$ -- **Equation 1**

Let us consider the value of m = log n i.e., $2^m = n$

⇨ $\sqrt{n} = n^{½} = 2^{m/2}$

Substitute the above values in equation 1.

$T(2^m) = 8\, T(2^{m/2}) + 1$ -- **Equation 2**

The above equation can be solved using the master's theorem

Let $S(m) = T(2^m)$

$S(\frac{m}{2}) = T(2^{m/2})$ -- **Equation 3**

Substituting the values of equation 3 in equation 2

\Theta

-3 $S(2^m) = 8\, S(\frac{m}{2}) + 1$

Using the master's method a = 8, b = 2, f(m) = 1= $m^0 = \theta(m^{\log_b a})$

Computing the value of $\log_b a = \log_2 8 = \log_2 2^3 = 3$

$\epsilon = ?$

The asymptotic bound

Case 1 applies

$$S(m) = \theta(m^{\log_b a}) = \theta(m^3)$$

Substituting the value of m = log n will give us

$$T(n) = \theta((\log n)^3)$$

Question 3:

A = (3, 2, 15, 9, 70, 18, 5, 33, 8)

Figure A: The 9 element array A and binary tree representation

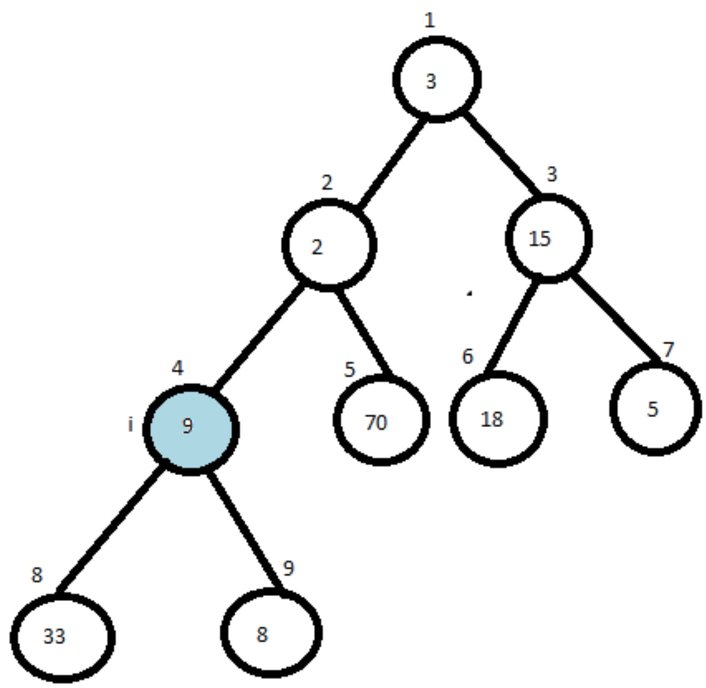


Figure B: The loop index i for the next iteration refers to node 3

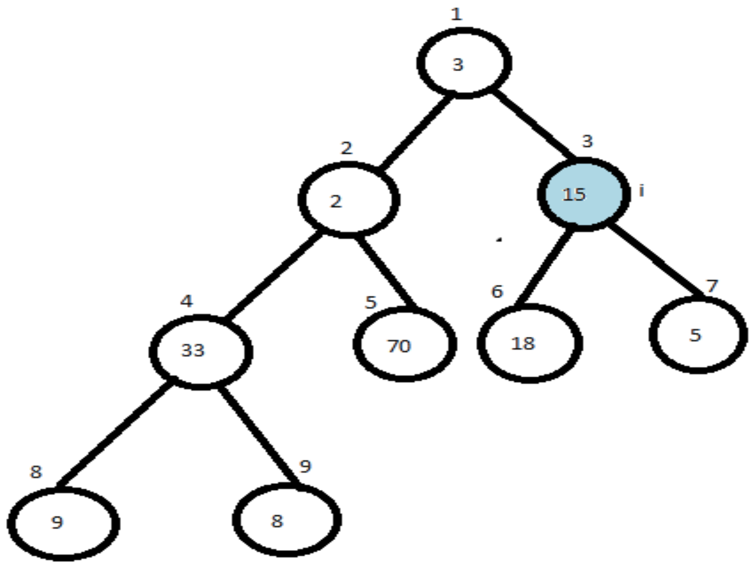


Figure C:

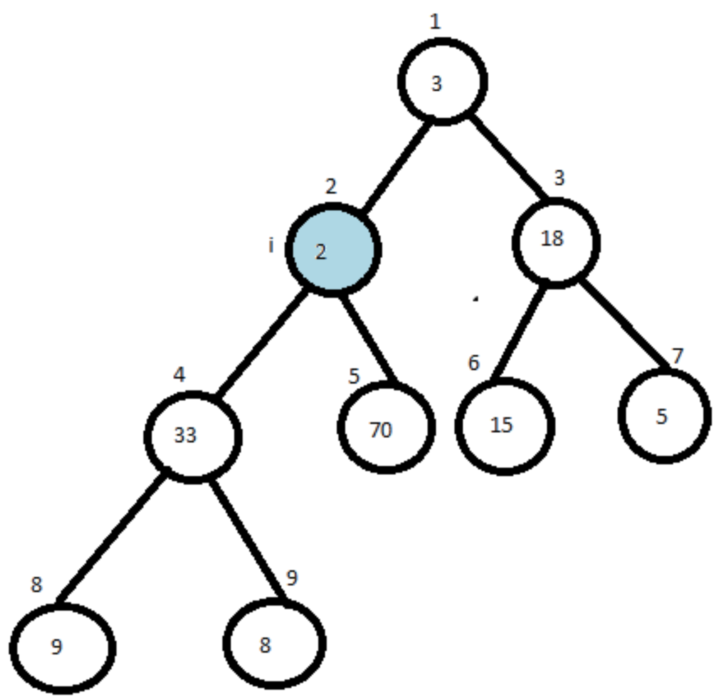


Figure D:

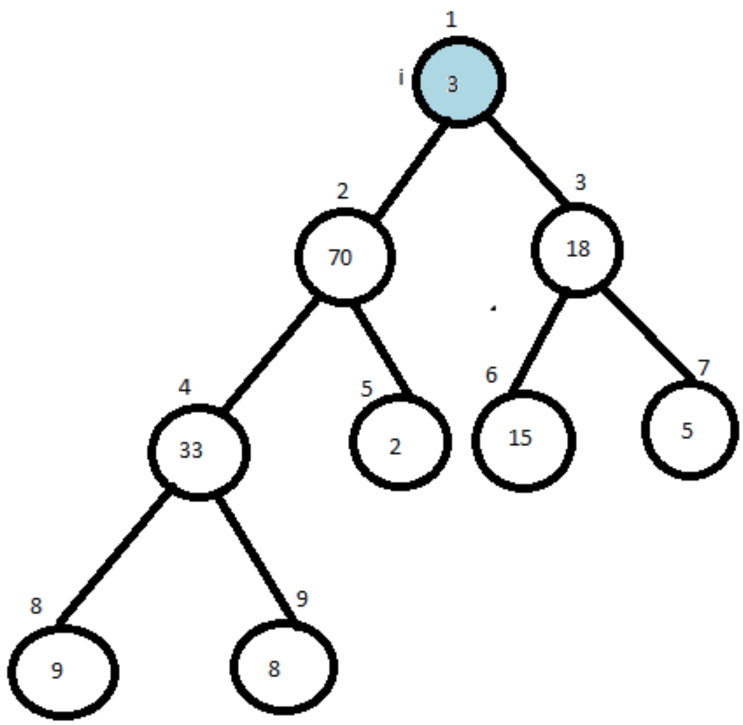


Figure E:

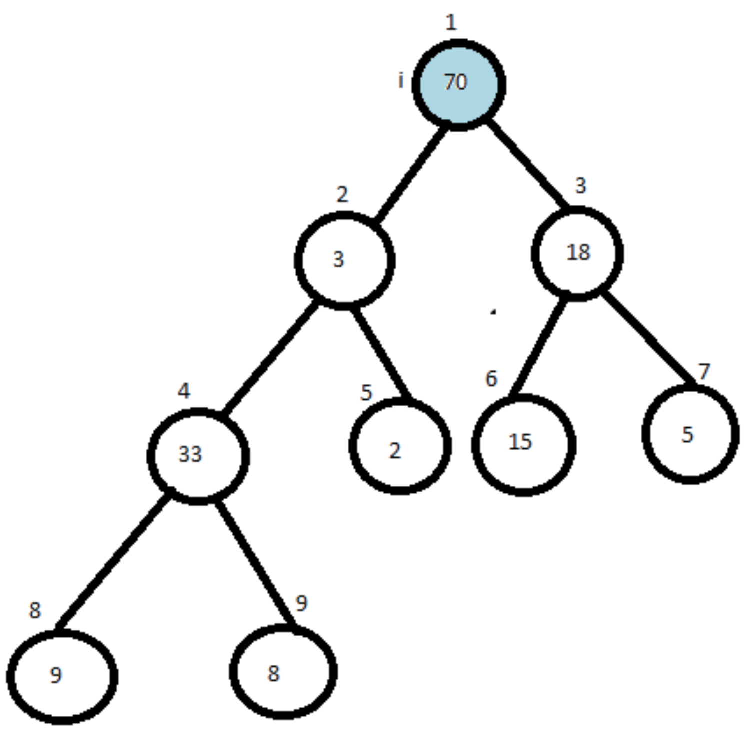
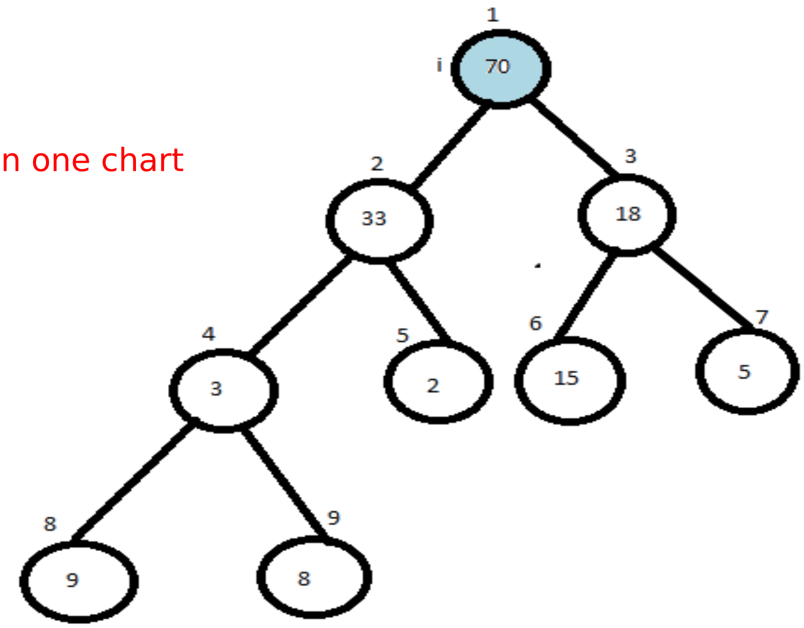


Figure F:

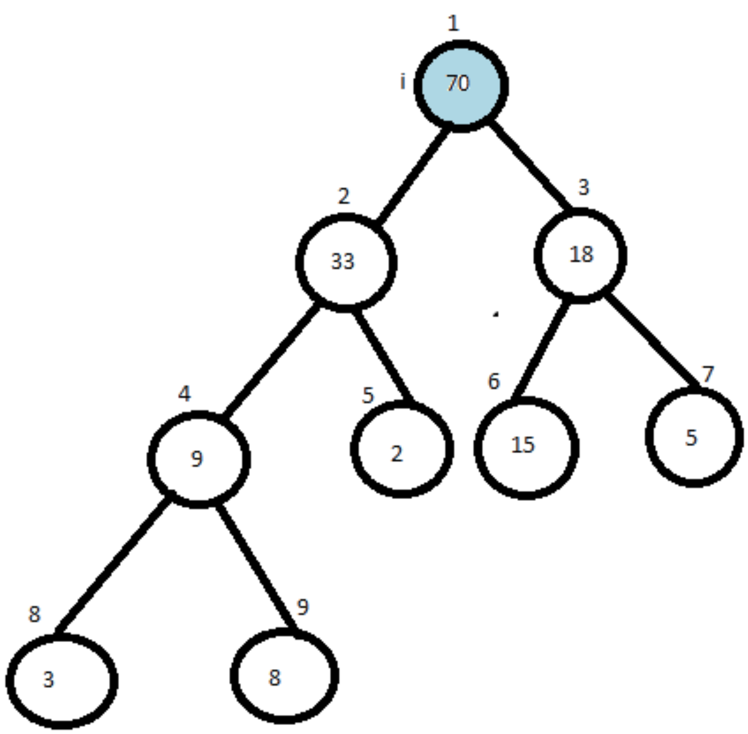
-1

Put E,F,G in one chart



Figures C-F: include subsequent iterations for the loop in BUILD-MAX-HEAP

Figure G: Max-Heap after the build max heap finishes



Question 4:

Algorithm:

```
OS_Select(x, i)
        r = x.left.size + 1
        if i == r
                return x
        else if i < r
                return OS_Select(x.left, i)
        else
                return OS_Select (x.right, i-r)
```

Figure A:

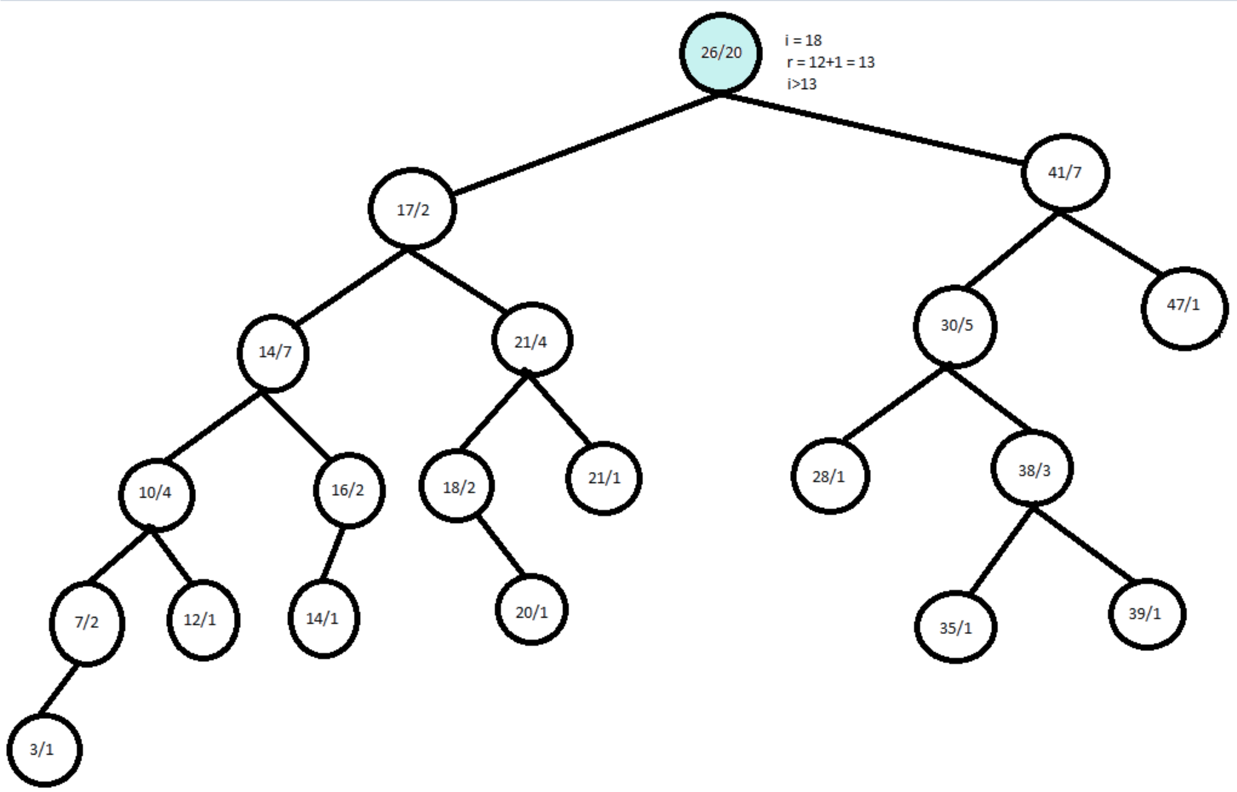


Figure B:

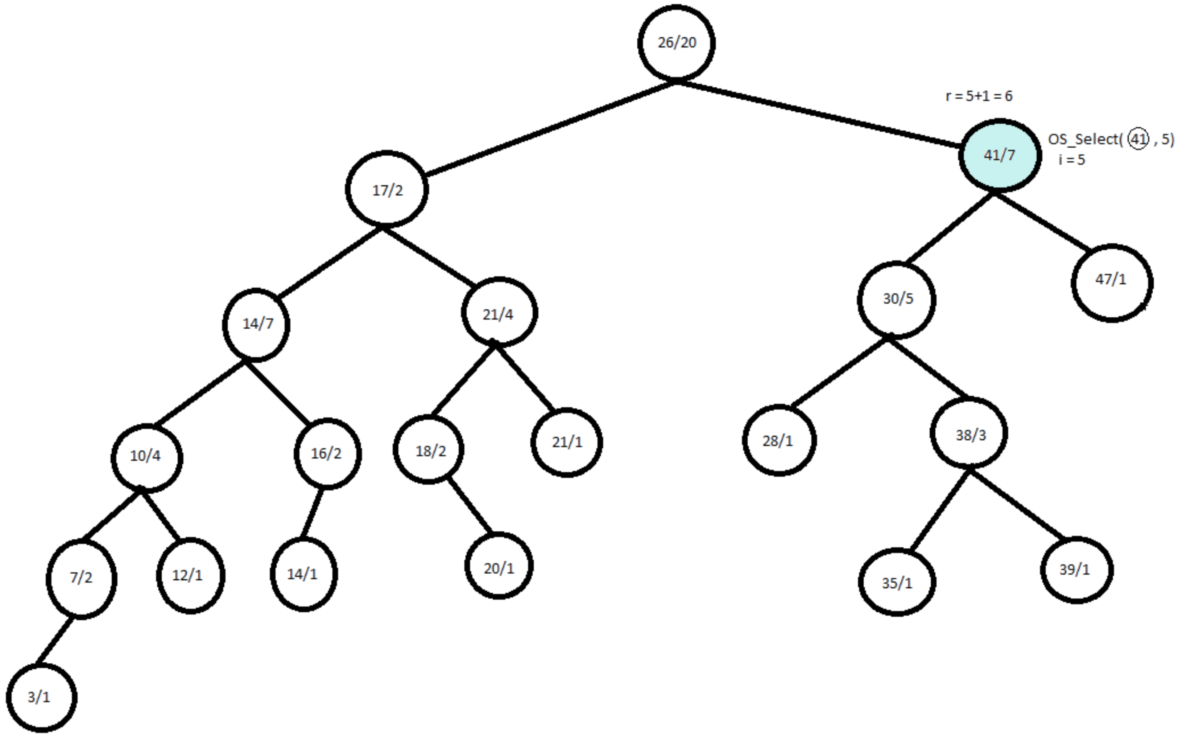


Figure C:

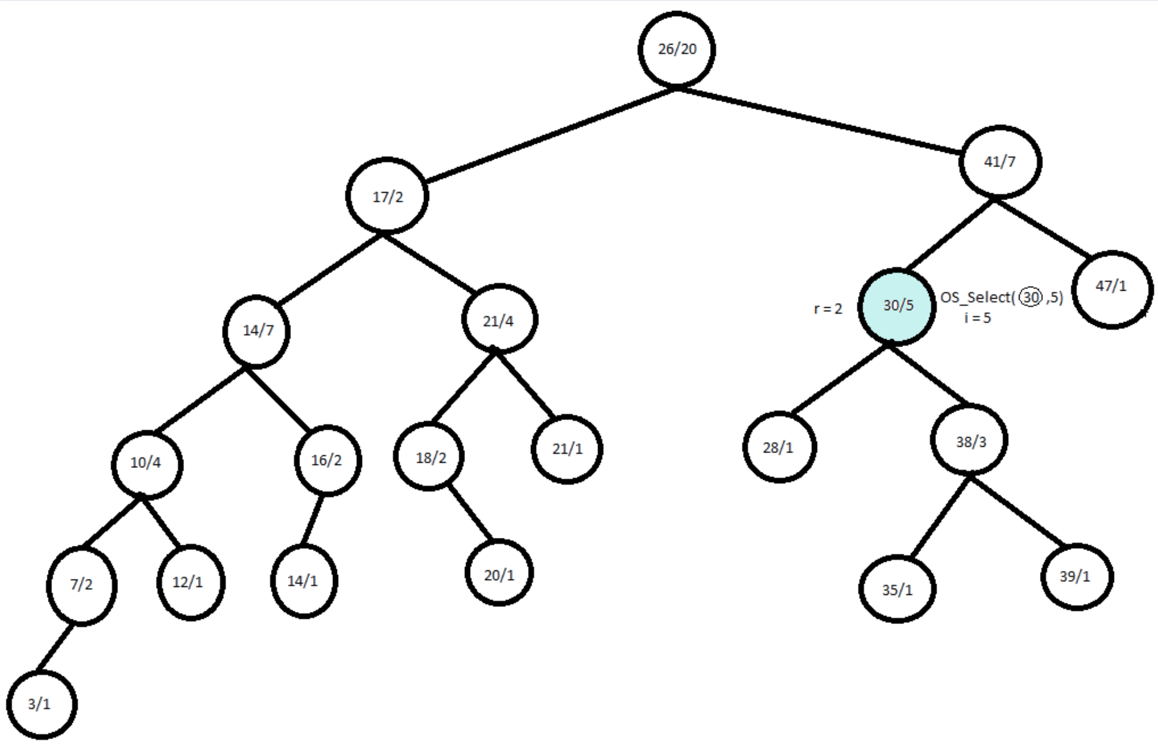


Figure D:

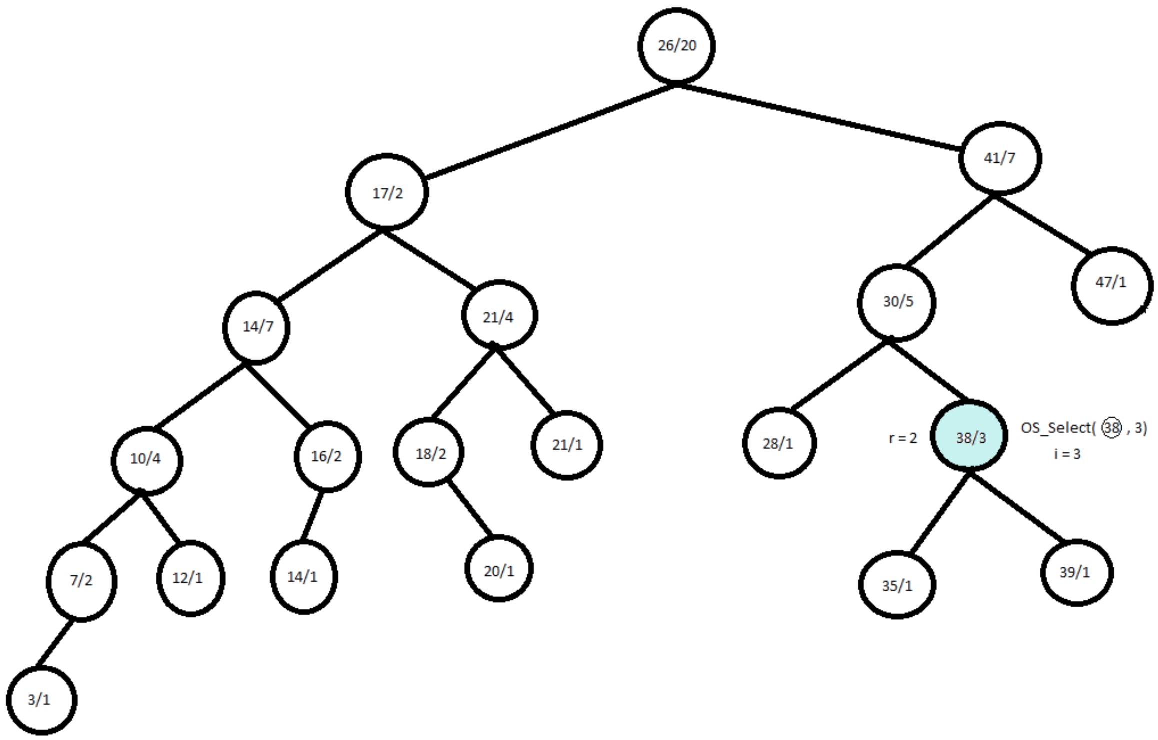
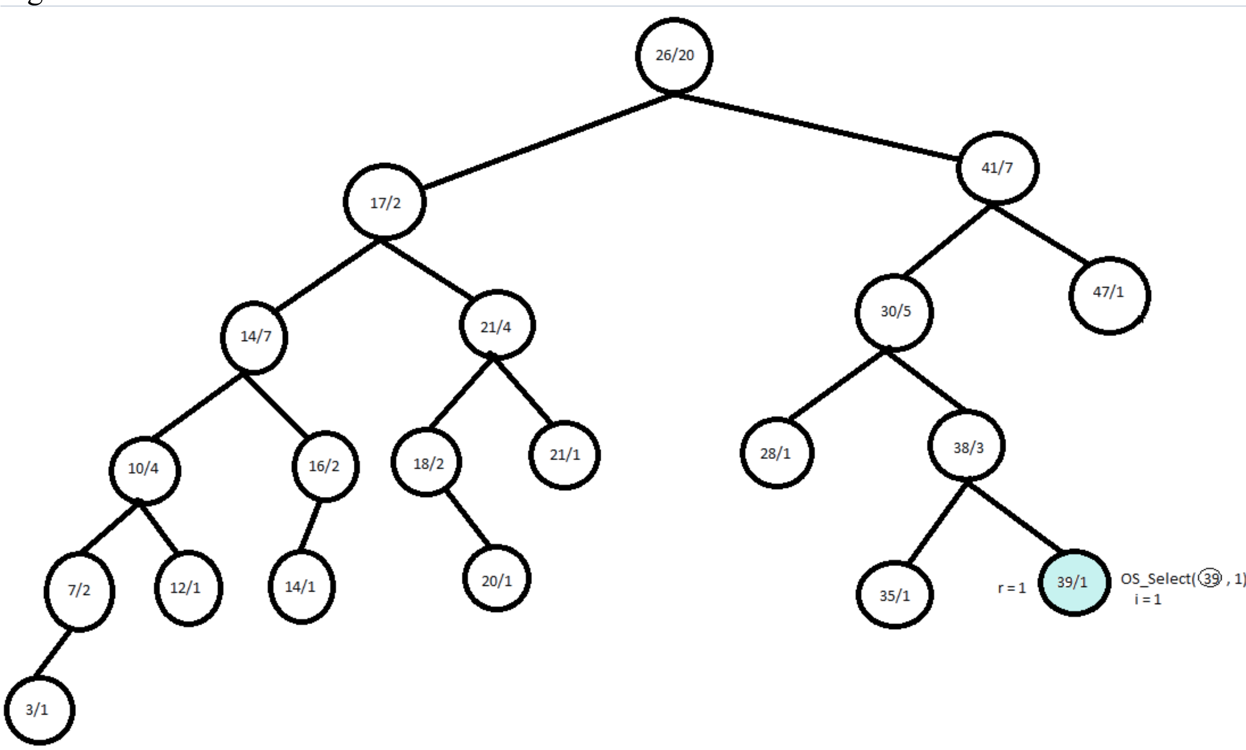
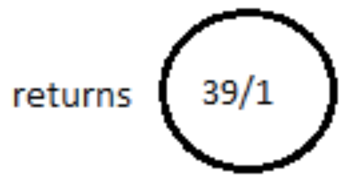

26/20
17/2
41/7
14/7
21/4
30/5
47/1
10/4
16/2
18/2
21/1
28/1
r = 2
38/3
OS_Select(38 , 3)
i = 3
7/2
12/1
14/1
20/1
35/1
39/1
3/1

Figure E:


26/20
17/2
41/7
14/7
21/4
30/5
47/1
10/4
16/2
18/2
21/1
28/1
38/3
7/2
12/1
14/1
20/1
35/1
r = 1
39/1
OS_Select(39 , 1)
i = 1
3/1



Question 5:

Given A = [1………. n] – Descending order

B = [1……… n] – Ascending Order

a)

BinSearch(A, l, h, v)

    if l >h

-23   Return NULL

  m = (l+h)/2

6) -15