P-1

CS-575 Homework 3 Amit Pandit
apandit1@binghamton.edu

Q1) input array = [10, 7, 3, 8, 1, 9, 0]
sort array using insertion sort

[10, 7, 3, 8, 1, 9, 0] A[i] > key
i j, j+1 = key as 10 > 7

[7, 10, 3, 8, 1, 9, 0] A[i] > key
i key as 10 > 3

[3, 7, 10, 8, 1, 9, 0] A[i] > key
i key as 10 > 8

[3, 7, 8, 10, 1, 9, 0] A[i] > key
i key as 10 > 1

[1, 3, 7, 8, 10, 9, 0] A[i] > key
i key as 10 > 9

[1, 3, 7, 8, 9, 10, 0] A[i] > key
i key as 10 > 0

[0, 1, 3, 7, 8, 9, 10]

$\boxed{\text{Sorted array} = [0, 1, 3, 7, 8, 9, 10]}$

Q2] input array = [13, 57, 39, 85, 70, 22, 64, 48]

sort using merge sort

13, 57, 39, 85, 70, 22, 64, 48

[13, 57, 39, 85] [70, 22, 64, 48]

[13, 57] [39, 85] [70, 22] [64, 48]

[13] [57] [39] [85] [70] [22] [64] [48]

[13, 39, 57, 85] [22, 48, 64, 70]

~~[13, 22, 48,~~

= [13, 22, 39, 48, 57, 64, 70, 85]

Sorted array → ~~[13, 3~~ [13, 22, 39, 48, 57, 64, 70, 85]

Q3] given $A = \begin{matrix} 2, 3 \\ 4, 5 \end{matrix}$ $B = \begin{matrix} 5, 6 \\ 1, 3 \end{matrix}$

$$\begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} = \begin{pmatrix} 2 & 3 \\ 4 & 5 \end{pmatrix} ; \begin{pmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{pmatrix} = \begin{pmatrix} 5 & 6 \\ 1 & 3 \end{pmatrix}$$

Creating matrix

$S_1 = B_{12} - B_{22} = 6 - 3 = 3$

$S_2 = A_{11} - A_{12} = 2 + 3 = 5$

$S_3 = A_{21} - A_{22} = 4 + 5 = 9$

P-3

$$S_4 = B_{21} - B_{11} = 1-5 = -4$$
$$S_5 = A_{11} + A_{22} = 2+5 = 7$$
$$S_6 = B_{11} + B_{22} = 5+3 = 8$$
$$S_7 = A_{12} + A_{22} = 3-5 = -2$$
$$S_8 = B_{21} + B_{22} = 1+3 = 4$$
$$S_9 = A_{11} - A_{21} = 2-4 = -2$$
$$S_{10} = B_{11} + B_{12} = 5+6 = 11$$

$$P_1 = A_{11} S_1 = 2\cdot 3 = 6$$
$$P_2 = S_2 B_{22} = 5\cdot 3 = 15$$
$$\cancel{P_3 = S_3 B_{11} = 5\cdot -4 = -20}$$
$$\cancel{P_4 = A_{22} S_4 = 7\cdot 8 = 56}$$
$$P_3 = S_3 B_{11} = 9\cdot 5 = 45$$
$$P_4 = A_{22} S_4 = 5\cdot(-4) = -20$$
$$P_5 = S_5 \cdot S_6 = 7\cdot 8 = 56$$
$$P_6 = S_7 \cdot S_8 = (-2)(4) = -8$$
$$P_7 = S_9 \cdot S_{10} = (-2)(11) = -22$$

$$C_{11} = P_5 + P_4 - P_2 + P_6 = 56 + (-20) - 15 + (-8)$$
$$= 56 - 43$$
$$= 13$$

$$C_{12} = P_1 + P_2 = 6 + 15 = 21$$
$$C_{21} = P_3 + P_4 = 45 - 20 = 25$$
$$C_{22} = P_5 + P_1 - P_3 - P_7 = 56 + 6 - 45 + 22$$
$$= 39$$

$\therefore$ Product of given matrices $= \begin{pmatrix} 13 & 21 \\ 25 & 39 \end{pmatrix}$

Q4) input array = [13, 9, 5, 7, 3, 1, 10, 6, 11, 2]

a) p, j ... r

i

13	9	5	7	3	1	10	6	11	2

p j ... r

i

13	9	5	7	3	1	10	6	11	2

p j ... r

i

13	9	5	7	3	1	10	6	11	2

p j

i

13	9	5	7	3	1	10	6	11	2

p j ... r

i

13	9	5	7	3	1	10	6	11	2

p j ... r

i

13	9	5	7	3	1	10	6	11	2

A[j] < r
∴ i = i+1
A[i] = A[j]

i,p j ... r

1	9	5	7	3	13	10	6	11	2

i,p

1	9	5	7	3	13	10	6	11	2

j r

1	9	5	7	3	13	10	6	11	2

i,p ... j r

exchange
A[i+1] with A[r]

1	2	5	7	3	13	10	6	11	9

b) Considering next sub array

1	2

i

5	7	3	13	10	6	11	9

p,j ... r

p - 5

5	7	3	13	10	6	11	9

p,i j r

$i = i+1$
$A[i] = A[j] \Rightarrow A[i] = 7$

5	7	3	13	10	6	11	9

p i j

$3 < 9$
$i = i+1$
$A[i] = A[j] \Rightarrow A[i] = 3$

5	7	3	13	10	6	11	9

p i j r

5	7	3	13	10	6	11	9

i j r

$6 > 9$
$i = i+1$
$A[i] = A[j] \Rightarrow A[i] = 6$

5	7	3	6	10	13	11	9

i j r

5	7	3	6	10	13	11	9

i j r

5 7 3 6 9 13 11 10

[5 7 3 6] [13 11 10]

r

[5 7 3 6] [10 11 13]

r

[5 3 7 6] [10 11 13]

[5, 3] [7] [11, 13]

[3, 5]

Q5] Sorted list of distinct integers in array
& we need to find index i such that $A[i]=i$ &
provide algorithm that runs in $O(\log n)$

For above scenario if the element search the element in right half of array if the element is less than middle element search the element in left half of array so we will basically use binary search for this

pseudo code:-

```
int binarysearch(int array[], int min, int max)
    if (max >= min)
        int mid = (max + min)/2

        if (array[mid] == mid)
            return mid
        else if (array[mid] < mid)
            return binary search(array, mid+1, max);
        else
            return binarysearch(array, min, mid-1)
```

Correctness:-
loop invariant:- An index i is in the array A[min...max] that is sorted in ascending order when length of array is greater than 1

initialization :-

it the array is of length 1 then element is present in the 1^{st} index only i.e $A[1]=1$ else it will return -1

maintainence :-

a] if $i \in A[mid+1, max]$ then i must be greater than middle element As the array is not changed it is still sorted and ~~wecu~~ ~~an~~ will recursively call binary search on(array, ~~&~~ mid+1, max) till array[mid] = mid

b] if $i \in A[min, mid] \Rightarrow i \leq [mid]$ i.e array is not changed and sorted & by recursively calling binary search on (array, min, mid-1) at one point array[mid] = mid

termination :-

if element is not present in array then it will return -1

if we calculate time complexity :-

time complexity :- Analyzing runtime of search function

The length of array is n, the recursion function is called it will get array/2 ie. $n/2$

So -- 1^{st} recursion is n

2^{nd} recursion is $n/2$

3^{rd} recursion is $n/4$

⋮

k^{th} recursion will be $n/2^k$

$$n/2^k = 1$$

$$\log_2 n = \log_2 2^k = k\log_2 2$$

Hence proved $\boxed{\text{Complexity is } O(\log n)}$

p - 8

Q6) probablity of getting 1 is $P(1) = \frac{1}{1}(x)$

where x is the Probablity

$P(2) = \frac{1}{2}(x)$ $\quad$ $P(3) = \frac{1}{3}(x)$

$P(4) = \frac{1}{4}(x)$ $\quad$ $P(5) = \frac{1}{5}(x)$

$P(6) = \frac{1}{6}(x)$

$$P(1) + P(2) + P(3) + P(4) + P(5) + P(6) = 1$$

$$\frac{1}{1} + \frac{1}{2}(x) + \frac{1}{3}(x) + \frac{1}{4}(x) + \frac{1}{5}(x) + \frac{1}{6}(x) = 1$$

$$\frac{60x + 30x + 20x + 15x + 12x + 10x}{60} = 1$$

$$\frac{147}{60} x = 1 \qquad \boxed{x = \frac{60}{147}}$$

probablity of number greater than 3 is

$$P(4) + P(5) + P(6) = \frac{1}{4}x + \frac{1}{5}x + \frac{1}{6}x$$

$$= x\left(\frac{1}{4} + \frac{1}{5} + \frac{1}{6}\right)$$

$$= \frac{60}{147}\left(\frac{1}{4} + \frac{1}{5} + \frac{1}{6}\right)$$

$= \cancel{x}\left(\right)$

$$= \frac{\cancel{60}}{147} \frac{(15 + 12 + 10)}{\cancel{60}}$$

$= x$

$$\boxed{= \frac{37}{147}}$$