Data Structure

Monday, June 19, 2023 7:49 AM

Section B - Data Structure - No. Of Questions - 10

- Searching
  - Linear Search
  - Binary Search
- Sorting
  - Selection Sort
  - Bubble Sort
  - Insertion Sort
  - Merge Sort
  - Quick Sort
- Stack
  - Implementation
  - Applications of stack - Expression conversion and evaluation
- Queue
  - Implementation Linear Queue Circular Queue
- Linked List
  - All four types - implementation
- Tree and Graph - Terminologies
- Hash Table

Data Structure:

Algorithm:

Logical flow of given problem statement

Pseudocode - can be represented using pseudocode or

Flowchart - logical flow represent in pictorial format

There are different algorithms are present:

1. Divide and Concur Algorithm
2. In place Algorithm
3. Greedy Algorithm

Program:

Set of instructions

Efficiency:

Can be decided based on timespan and memory consumed by a specific algorithm.

Timespan can be calculated using concept Time Complexity

Memory consumed by any algorithm can be calculated using a concept of Space Complexity

Time Complexity is measured in 2 different ways

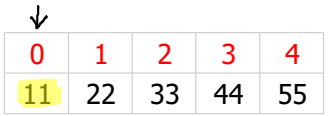
1. Asymptotic Time Complexity
    1. Best Case - Omega notation
    2. Average Case - Thita notation
    3. Worst Case - Big Oh notation
2. Symptotic Time Complexity

Space Complexity is calculated based on 2

factors

1. Fixed Space
2. Variable Space

Monday, June 19, 2023 8:27 AM



0	1	2	3	4
11	22	33	44	55

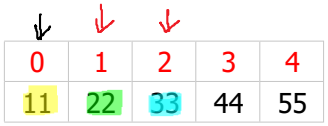
Key = 11

a[0] == 11

comparision count = 1

best time

O(1)



0	1	2	3	4
11	22	33	44	55

Key = 33

a[0] == 33 ×

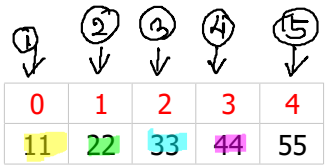
a[1] == 33 ×

a[2] == 33 ✓

comparision count = 3

Average.

(n+1)/2



0	1	2	3	4
11	22	33	44	55

Key = 55

a[0] == 55 ×

a[1] == 55 ×

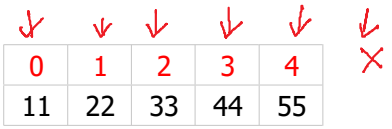
a[2] == 55 ×

a[3] == 55 ×

a[4] == 55 ✓

comparision count = 5

O(n)



0	1	2	3	4
11	22	33	44	55

Key = 28

a[0] == 28 ×

a[1] == 28 ×

a[2] == 28 ×

a[3] == 28 ×

a[4] == 28 ×

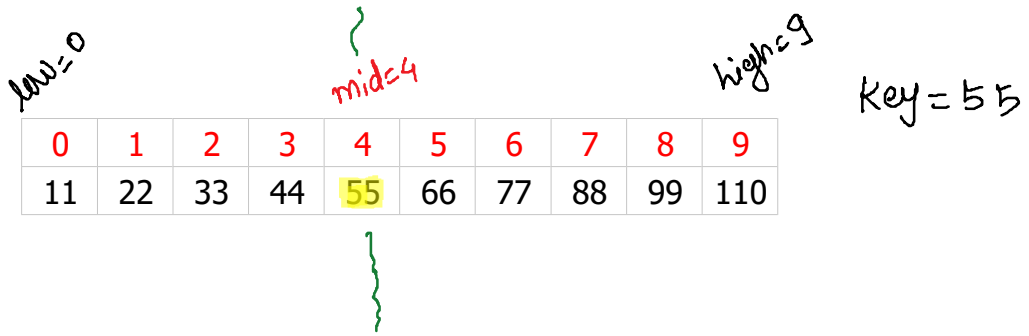
Binary Search uses divide and concur algorithm

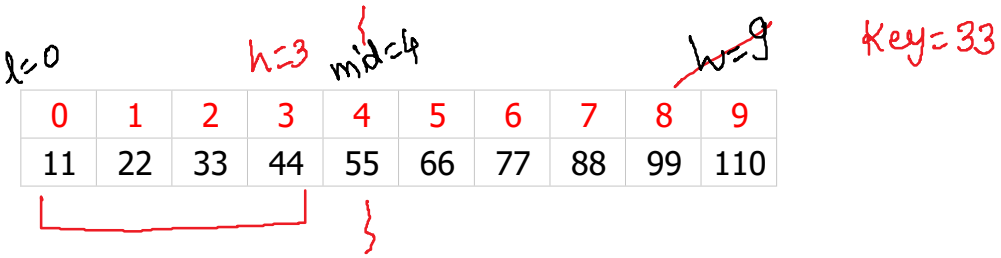
One of the fastest searching algorithm

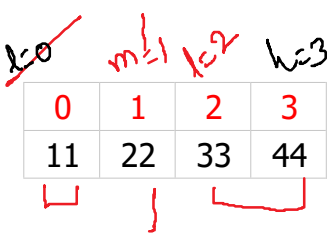
Pre-requisite of binary search algorithm is data has to be in sorted order

If key is present in collection it will be always at mid location

Divide and Concur algorithm is used by
1. Binary Search
2. Merge Sort
3. Quick Sort



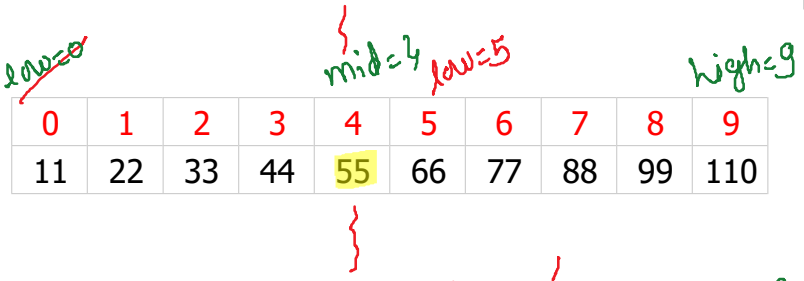


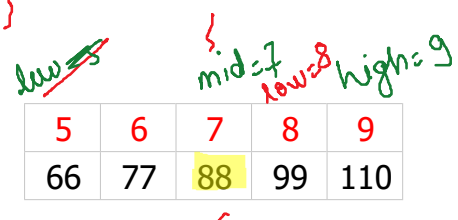


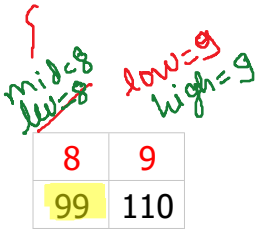

```
int low=0, high=9, mid


while(low<=high)
{
mid = (low+high)/2;
if(a[mid] == key)
    return mid;
else if(key < a[mid])
    high = mid -1;
else if(key > a[mid])
    low = mid + 1;
}
```

2	3
33	44

key=110








```
int low=0, high=9, mid


while(low<=high)
{
mid = (low+high)/2;
if(a[mid] == key)
    return mid;
else if(key < a[mid])
    high = mid -1;
else if(key > a[mid])
    low = mid + 1;
}
```

Monday, June 19, 2023 9:25 AM

In case of selection sort lowest element of collection is placed in order after completion of first pass

0	1	2	3	4
33	22	11	25	10

Selection sort algorithm needs to execute n-1 passes

1st pass ==== n-1 comparisons == 5-1 = 4
2nd pass ==== n-2 comparisons== 5-2 = 3
3rd pass ==== n-3 comparisons== 5-3 = 2
4th pass ==== n-4 comparisons== 5-4 = 1

Given n elements if we apply Selection Sort/ BubbleSort/MergeSort/Insertion Sort/ Quick sort then after completion of specific pass what will be state of an collection

```
for(i=0; i < size-1 ; i++)        pass. select element
    {
      for(j=i+1 ; j <size ; j++)   compare with remaining elements
      {
          if(p[i] > p[j])
          {
              p[i] = p[i] + p[j];   a=a+b
              p[j] = p[i] - p[j];   b=a-b
              p[i] = p[i] - p[j];   a=a-b
          }
      }
    }
```

a = 5 b = 7
a = 5+7 = a=12
b = 12-7 = b=5
a = 12-5 = a=7

i=0 j=1

0	1	2	3	4
33	22	11	25	10

i<4 ✓
j<5 ✓

0	1	2	3	4
22	33	11	25	10

i=0 j=2

0	1	2	3	4
22	33	11	25	10

2<5 ✓

0	1	2	3	4
11	33	22	25	10

i=0 j=3

0	1	2	3	4
11	33	22	25	10

3<5 ✓

i=0 j=4

0	1	2	3	4
11	33	22	25	10

4<5

j=5

0	1	2	3	4
10	33	22	25	11

5<5

i=1 j=2

0	1	2	3	4
10	33	22	25	11

1<5-1
2<5 ✓

i=1 j=3

0	1	2	3	4
10	22	33	25	11

3<5 ✓

i=1 j=4

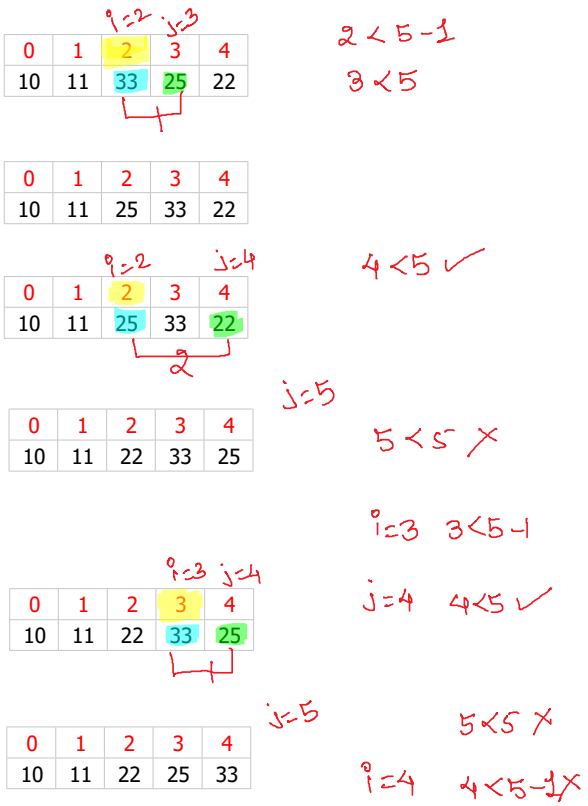
0	1	2	3	4
10	22	33	25	11

4<5 ✓

j=5

0	1	2	3	4
10	11	33	25	22

5<5 X



0	1	2	3	4
10	11	33	25	22

0	1	2	3	4
10	11	25	33	22

0	1	2	3	4
10	11	25	33	22

0	1	2	3	4
10	11	22	33	25

0	1	2	3	4
10	11	22	33	25

0	1	2	3	4
10	11	22	25	33

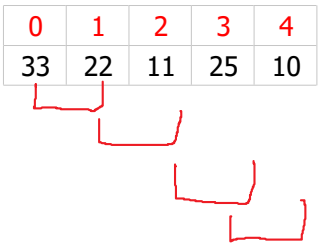
Monday, June 19, 2023 10:12 AM

In case of bubble sort highest element from collection will be place in order

Bubble sort algorithm executes n-1 passes

Time Complexity = (n-1) + (n-2)+n-3) …

```
int i,j;
for(i=0;i<size-1;i++)
{
    for(j=0;j<size-1-i;j++)
    {
        if(p[j] > p[j+1])
        {
            p[j] = p[j] + p[j+1];
            p[j+1] = p[j] - p[j+1];
            p[j] = p[j] - p[j+1];
        }
    }
}
```

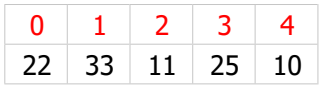


i=0 1st pass

0 < 5-1

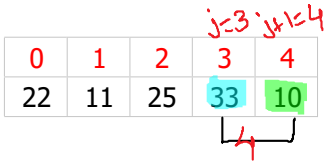
j < 5-1-i

0 < 5-1-0 ✓

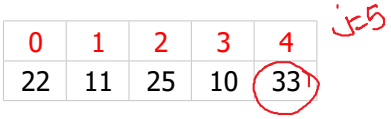


1 < 5-1-0 ✓

2 < 5-1-0 ✓

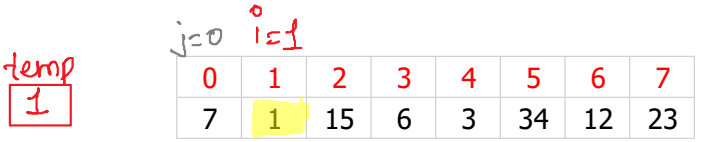


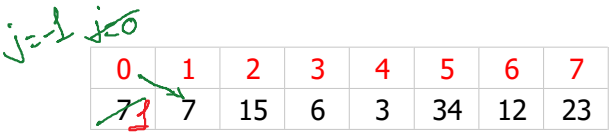
3 < 5-1-0 ✓

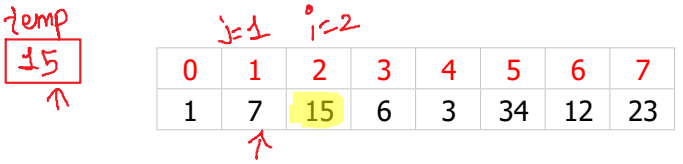


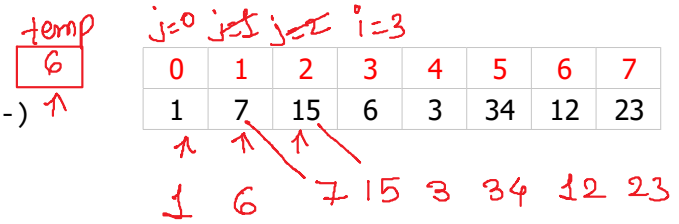
5 < 5-1-0 ✗

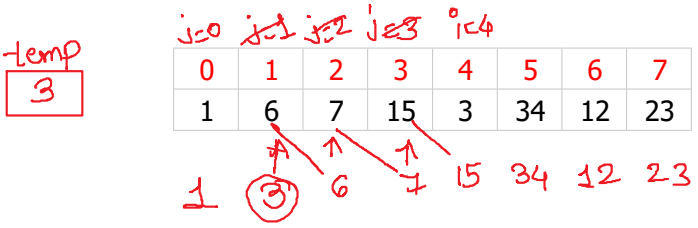
```
int i, j;
    int temp;
    for(i=1; i<n; i++)
    {
      temp = arr[i];
       for(j=i-1; j>=0 && arr[j] > temp; j--)
            arr[j+1] = arr[j];
      arr[j+1] = temp;
    }
```

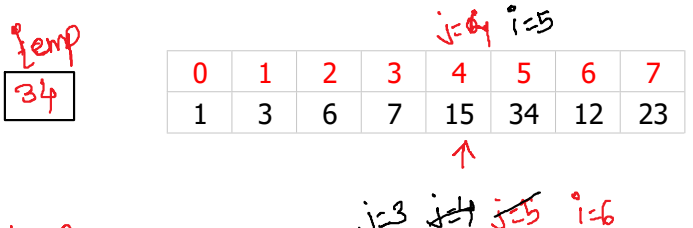


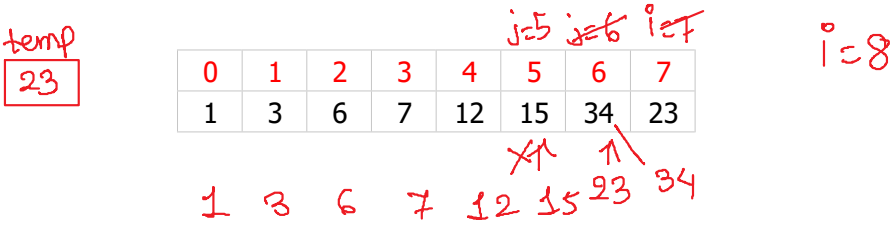


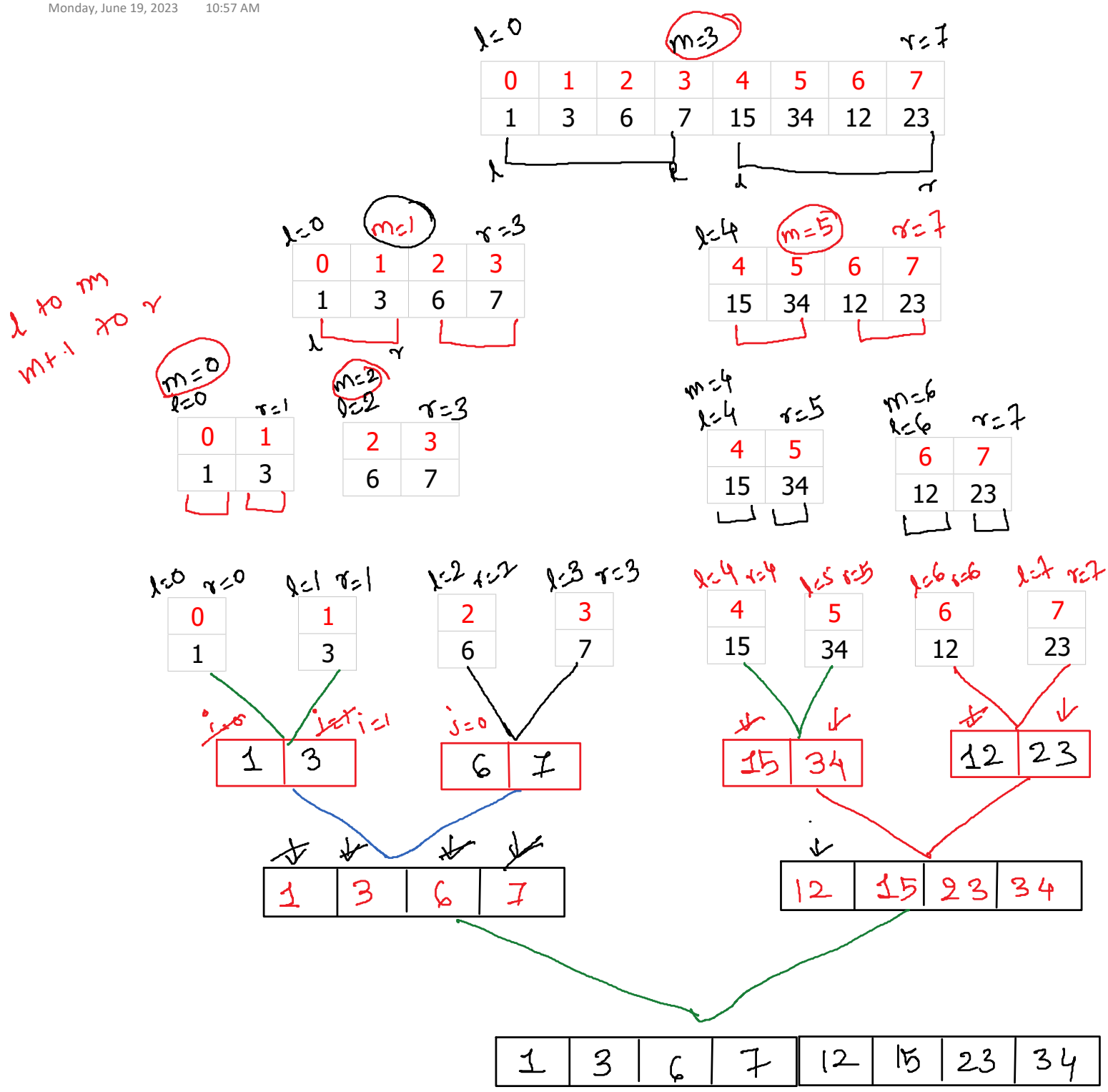











l=0
m=3
r=7
0 1 2 3 4 5 6 7
1 3 6 7 15 34 12 23
l to m
m+1 to r
l=0
m=1
r=3
0 1 2 3
1 3 6 7
l=4
m=5
r=7
4 5 6 7
15 34 12 23
m=0
l=0
r=1
0 1
1 3
m=2
l=2
r=3
2 3
6 7
m=4
l=4
r=5
4 5
15 34
m=6
l=6
r=7
6 7
12 23
l=0 r=0
l=1 r=1
l=2 r=2
l=3 r=3
l=4 r=4
l=5 r=5
l=6 r=6
l=7 r=7
0
1
1
3
2
6
3
7
4
15
5
34
6
12
7
23
i=0
j=i+1
i=1
j=0
1 3
6 7
15 34
12 23
1 3 6 7
12 15 23 34
1 3 6 7 12 15 23 34

Monday, June 19, 2023 10:57 AM