Data Structure & Algorithms

Sunbeam Infotech

Agenda

- Q & A
- Linear Search
- Binary Search
- Selection Sort
- Bubble Sort
- Insertion Sort
- Binary Search (recursive)

Trainer Introduction

- Name: Mr. Nilesh Ghule.
- Qualification: M.Sc. Electronics
- Experience: 16+ years training along with consulting & POC for Sunbeam.
- Designation: Technical Director of Sunbeam Infotech.

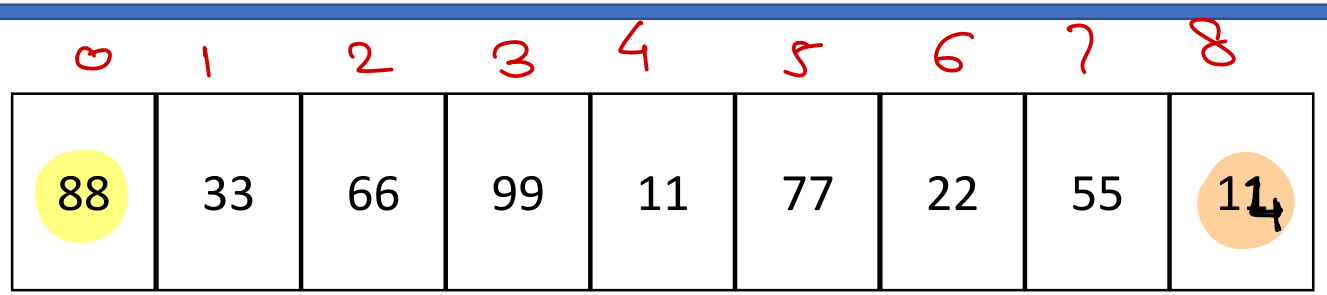
- Skills/Technologies/Platform:
  - C, C++, Java, JS, Scala, Python, C#.
  - Java EE (Spring, Hibernate, Spring Boot).
  - Operating Systems, Linux, Device Drivers, RTOS
  - MySQL, MongoDb, Redis, HBase, Neo4J.
  - Big Data, Hadoop & Eco Systems, Spark, Kafka
  - ARM7, ARM-CM3, AVR, 8051 & IoT
  - Win32 SDK, MFC, COM, .NET
  - Symbian, J2ME, Windows CE, Android.
- Contact:
  - nilesh@sunbeaminfo.com
  - gitlab: nilesh-g
  - linkedin.

Course Introduction

- Data Structure and Algorithms
  - Data Structures: Linked lists, Stack, Queue, Trees, Graphs, Hash Table.
  - Algorithms: Sorting, Searching, Graphs & Other DS algorithms
- Course Goals
  - Learn DS & Algo from scratch.
  - Implement each DS & Algo.
  - Understand complexity of algos.
- Course Schedules
  - Mon-Fri: 6:30 PM to 9:30 PM
  - Sat: 6:30 PM to 7:30 PM
- Course Format
  - 6:30 PM to 9:30 PM – Online lecture
  - Participants are encouraged to code alongside (at least copy the code from gitlab).
  - Post your queries in chat box.
  - Practice assignments will be shared. They are optional. If any doubts, share on WA group (possibly with screenshot). Faculty members or peers can help.
- Programming language
  - DS & Algos are language independent.
  - Classroom coding will be in C++ (use IDE of your choice).
  - Will share Java codes at the end of session.

Linear Search

- Find a number in a list of given numbers (random order).



- Time complexity
  - Worst case → last ele search
    itr = n T α n → O(n)
  - Best case → first ele search
    itr = 1 T = k → O(1)
  - Average case → random ele search
    itr = n/2 (avg) T α n/2 → O(n)

```
int lin_search(int a[], int n, int key)
{
    for(i=0; i<n; i++)
    {
        if(a[i]==key)
            return i;
    }
    return -1;
}
```

Binary Search → only work with sorted array.

```
l = 0
r = n - 1
while (l <= r)
{ m = (l + r) / 2
  if (key == a[m])
      return m;
  if (key < a[m])
      r = m - 1;
  else // key > a[m]
      l = m + 1;
}
```

0	1	2	3	4	5	6	7	8
11	22	33	44	55	66	77	88	99

$2^{itr} = n$

r l → m → ele not found

$\log 2^{itr} = \log n$

$itr \log 2 = \log n$

$itr = \frac{\log n}{\log 2}$ → $T \propto \frac{\log n}{\log 2}$

$T \propto \log n$

$O(\log n)$

Binary Search

5

m = (0+8)/2 = 4

m = (0+3)/2 = 1

m = (0+0)/2 = 0

-1

0	1	2	3	4	5	6	7	8
11	22	33	44	55	66	77	88	99

r < l

m

$2^4 = 16$

sorting algorithms

1. selection
2. bubble
3. insertion

---

4. quick
5. merge

---

6. heap

Selection Sort

Select ith element ($i \rightarrow 0$ to $n-1$)

Comp with all ele after that (j)

```
if(a[i] > a[j])
    Swap(a[i], a[j])
```

0	1	2	3	4	5
11	22	33	44	55	66

↑ i ↑ j

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

i

0 → 5
1 → 4
2 → 3
3 → 2
4 → 1

15

num of itrs

$= 1+2+3+\dots+(n-1)$

itr $= \frac{n(n-1)}{2}$

$T \propto n^2 - n$

$T \propto n^2$ → $O(n^2)$

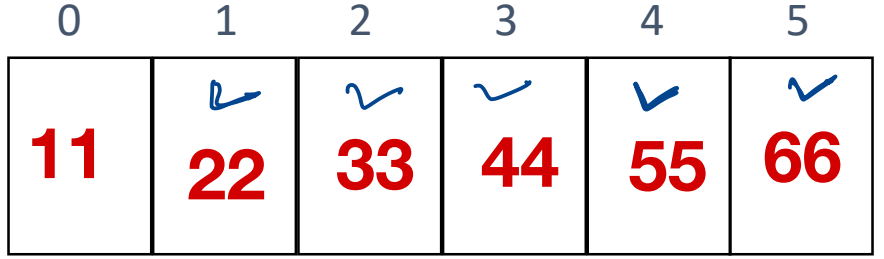
Bubble Sort

repeat for n-1 times.

Compare two consecutive eles.

if left ele > right ele,

swap them.




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

i → 5

2 → 5

3 → 5

4 → 5

5 → 5

(25)

itr = $(n-1)^2$

$T \propto (n-1)^2$

$T \propto n^2 - 2n + 1$

$T \propto n^2$

$O(n^2)$

Bubble Sort

i = 1

i = 2

i = 3

i = 4 → no swapping done

↓

implies → array is sorted.

↓

no further passes needed.

0	1	2	3	4	5
11	22	33	44 ✓	55 ✓	66 ✓

Thank you!

Nilesh Ghule <nilesh@sunbeaminfo.com>