

3/12/2015

(BT- Brum Adamin Roman)

= Arrays =

main

```
int [] midterm = new int[50];
```

```
midterm[0] = 100;
```

```
midterm[1] = 80;
```

```
int avg = (midterm[0] + midterm[1]) / 2;
```

```
public static float avgScore (int[] testScore)
```

```
{
```

```
float sum = 0.0;
```

```
for (int i = 0; i < testScore.length; i++)
```

```
{  
    sum = sum + testScore[i];  
}
```

```
return sum;
```

```
}
```

Sorting Searching:

Donald Knuth 3rd Vol. Sorting & searching

Sorting:

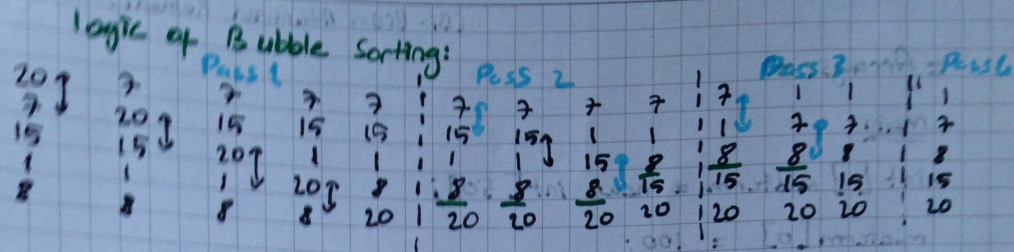
Bubble Sort

Selection sort
Insertion

Merge sort

Redix sort

Quick Sort.



Sort in increasing order = $1 \leq \text{No. of Pass} \leq n-1$

Ascending order: # of comparisons = $4 + 3 + 2 + 1$

non-decreasing order = $\leftarrow \text{unsorted} \rightarrow$
 $\leftarrow \text{sorted} \rightarrow$

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

$$\text{Total comparisons} = \frac{n(n-1)}{2}$$

$$\frac{n^2}{2} - \frac{n}{2} \text{ and } n^2 \text{ and } O(n^2)$$

cost of algorithm

Array List:

`ArrayList<String> friends = new ArrayList<String>();` // declaration

`friends.add("Alper")`

`String name = friends.get(1);`

`import java.util.ArrayList;`

`int r = friends.size();`

`friend.add(1, "OYA");`

- 0 Alper
- 1 OYA
- 2 Tekin
- 3 Ali

1st Printing:

`friends.remove(1);`

`System.out.print(friends);`

2nd Printing: / Enhanced for loop

`for (String name : friends)`

`{ System.out.println(name);`

`}`

3rd: you can put in for loop

`for (int i = 0; i < friends.size(); i++)`

37