

Day:

Date:

Submitted by:

Khansa bint-e-Tariq

Subject

Programming Fundamental

## Assignment

Submitted to :

Sir Waseem Khan

Baba Guru Nanak University  
Nankana Sahib

BSCSF24MDA

## Random Sorting

Q: [11, 1, 5, 4, 3, 2]

[5, 4, 2, 11, 3, 1] → not sorted

[3, 1, 2, 4, 5, 11] → not sorted

[1, 2, 3, 4, 5, 11] → sorted

A: [18, 12, 9, 20, 15, 10, 8]

[15, 8, 9, 20, 10, 18, 12] → not sorted

[10, 9, 12, 15, 20, 18, 8] → not sorted

[8, 9, 10, 12, 15, 18, 20] → sorted

A: [35, 38, 32, 39, 31, 33, 36, 40]

[32, 31, 40, 35, 33, 37, 36, 38] → not sorted

[33, 32, 31, 38, 39, 40, 35, 36] → not sorted

[31, 32, 33, 35, 36, 38, 39, 40] → sorted

A: [50, 46, 35, 28, 65, 76, 9, 18, 42]

[35, 65, 42, 9, 76, 50, 28, 18, 46] → not sorted

[9, 18, 28, 35, 46, 50, 42, 76, 65] → not sorted

[9, 18, 28, 35, 42, 46, 50, 65, 76] → sorted

# Quick Sorting

A: [11, 1, 5, 4, 3, 2]

Partitioning :-

11 > 2 → leave it to right

1 < 2 → move it to left

5 < 2 → leave it on right

4 > 2 → leave it on right

3 > 2 → leave it on right

[1, 2, 5, 4, 3, 11]

Pivot index 1

left subarray : [1]

right subarray [5, 4, 3, 2]

Pivot : 11

5 < 11 leave it on left

4 < 11 leave it on left

3 < 11 leave it on left

2 < 11 leave it on left

[5, 4, 3, 11]

Pivot index 3

left subarrays [5, 4, 3]

5 > 3 leave it on right

4 > 3 leave it on right

[3, 4, 5]

Pivot index : 0

Right subarray : [4, 5]

Pivot : 5

4 < 5 leave it on left

itself [4, 5]  
Pivot index 1

left part : [1]

Middle pivot : [2]

Sorted right part : [3, 4, 5, 11]

Final sorted list : [1, 2, 3, 4, 5, 11]

Ques:

[18, 12, 9, 20, 15, 10, 8]

Pivot : 8

$18 > 8 \rightarrow$  leave it on right

$12 > 8 \rightarrow$  leave it on right

$9 > 8 \rightarrow$  leave it on right

$20 > 8 \rightarrow$  leave it on right

$15 > 8 \rightarrow$  leave it on right

$10 > 8 \rightarrow$  leave it on right

[8, 12, 9, 20, 15, 10, 18]

Pivot index : 0

left subarray : [ ]

right subarray : [12, 9, 20, 15, 10, 18]

Pivot : 18

$12 < 18 \rightarrow$  leave it on left

$9 < 18 \rightarrow$  leave it on left

$20 > 18 \rightarrow$  leave it on right

$15 < 18 \rightarrow$  leave it on left

$10 < 18 \rightarrow$  leave it on left

Pivot index : 4

left subarray : [12, 9, 15, 10]

Pivot: 10

$12 > 10 \rightarrow$  leave it on right

$9 < 10 \rightarrow$  leave it on left

$15 > 10 \rightarrow$  leave it on left

$[9, 10, 15, 12]$

Pivot : 1

Right subarray  $(15, 12)$

Pivot : 12

$15 > 12 \rightarrow$  leave it on right

$[12, 15]$

Pivot : 2

left subarray :  $(8)$

sorted left subarray :  $[9, 10, 12, 15]$

Pivot : 18

Right part :  $(20)$

$[8, 9, 10, 15, 18, 20]$

Ques:-

$(35, 38, 32, 39, 31, 33, 36, 40)$

Pivot : 40

$35 < 40 \rightarrow$  leave it on left

$38 < 40 \rightarrow$  leave it on left

$32 < 40 \rightarrow$  leave it on left

$39 < 40 \rightarrow$  leave it on left

$31 < 40 \rightarrow$  leave it on left

$33 < 40 \rightarrow$  leave it on left

$36 < 40 \rightarrow$  leave it on left

(35, 38, 32, 39, 31, 33, 36, 40)

Pivot index : 7

left subarray : (35, 38, 32, 39, 31, 33, 36)

Pivot : 36

$35 < 36 \rightarrow$  leave it on left

$38 > 36 \rightarrow$  leave it on right

$32 < 36 \rightarrow$  leave it on left

$39 > 36 \rightarrow$  leave it on right

$31 < 36 \rightarrow$  leave it on left

$33 < 36 \rightarrow$  leave it on left

right subarray : (39, 38)

$39 > 38 \rightarrow$  leave it to right

Pivot index : 5

left subarray : (35, 32, 31, 33)

Pivot : 33

$35 > 33 \rightarrow$  leave it on right

$32 < 33 \rightarrow$  leave it on left

$31 < 33 \rightarrow$  leave it on left

Pivot index : 2

left subarray of (32, 31)

Compare  $32 > 31 \rightarrow$  leave it on right

Pivot : 33

Right part : (35)

Combine with (36, 38, 39, 40)

(31, 32, 33, 35, 36, 38, 39, 40)

Ques:-

[35, 38, 32, 39, 31, 33, 36, 40]

divide array into

Merge sorting

[35, 38, 32, 39] & [31, 33, 36, 40] halves

[35, 38], [32, 39] & [31, 33], [36, 40]

[35], [38], [32], [39], & [31], [33], [36], [40]

[35, 38], [32, 39] subarray merge

Compare 35 → 32 Add 32

Compare 35 → 39 Add 35

Compare 38 → 39 Add 38

Remaining 39

[32, 35, 38, 39]

[31, 33], [36, 40]

Compare 31 → 36 Add 31

Compare 33 → 36 add 33

Compare 36 → 40 Add 36

Remaining 40

[31, 33, 36, 40]

Compare 32 → 31 Add 31

Compare 32 → 33 Add 32

Compare 35 → 33 Add 33

Compare 35 → 36 Add 35

Compare 38 → 36 Add 36

Compare 38 → 39 Add 38

Compare 39 → 40 Add 39

Add remaining 40

[31, 32, 33, 35, 36, 38, 39, 40]

a = [18, 12, 9, 20, 15, 10, 8]

### Merge sorting

[18, 12, 9] & [20, 15, 10, 8]

[18, 12], [9] & [20, 15], [10, 8]  
[12, 18]

[12, 18] & [9] (9)

9 → 12 [9, 12]

9 → 18 (9, 12, 18)  
Remaining 18

[20], [15] and [10], [8]

Merge [15, 20] & [8, 10]

Merge [8, 15, 20] & [10]

8 → 10 [8, 10, 15, 20]

Sorted merged arrays :-

Compare 9 and 8 → Add 8

Compare 9 and 10 → Add 9

Compare 12 and 10 → Add 10

Compare 12 and 15 → Add 12

Compare 18 and 15 → Add 15

Compare 18 and 20 → Add 18

Add remaining 20

[8, 9, 10, 12, 15, 18, 20].

Q. NO. 01

[11, 1, 5, 4, 3, 2]

Merge sorting

[11, 1, 5] & [4, 3, 2]

Divide array  
into 2 halves.

[11], [1], [5] & [4], [3], [2]

split for.

[1, 5] Compare 1 → 5

Subarray merge

[1, 5]

[11] & [1, 5] [1]

11 → 01 [1, 5]

11 → 5 [1, 5, 11]

Remaining 11

Merge [3] & [2]

[2, 3]

Compare 2 → 3 [2, 3]

Merge [4], [2, 3]

4 → 2 → Add 2

4 → 3 → Add 3 [2, 3, 4]

Remaining 4 → Add 4

Sort merged arrays :-

[1, 5, 11] and [2, 3, 4]

Compare 1 → 2 Add 1

Compare 5 → 2 Add 2

Compare 5 → 3 Add 3

Remaining 5, 11 1, 2, 3, 4, 5, 11

## Insertion sorting

$(18, 12, 9, 20, 15, 10, 8)$

$\underline{18, 12, 9, 20, 15, 10, 8}$  u.s      current 12

$\underline{12, 18, 9, 20, 15, 10, 8}$  u.s      current 12

$\underline{9, 12, 18, 20, 15, 10, 8}$  u.s      current 20

$\underline{9, 12, 18, 20, 15, 10, 8}$  u.s      current 15

$\underline{9, 12, 15, 18, 20, 10, 8}$  u.s      current 17

$\underline{9, 10, 12, 15, 18, 20, 8}$  u.s      current 8

$8, 9, 10, 12, 15, 18, 20 \rightarrow$  sorted.

## Selection sort

$(18, 12, 9, 15, 10, 8)$  changing 8 with 18  
u.s

$\underline{8, 12, 9, 15, 10, 18}$  changing 9 with 12

$\underline{8, 9, 12, 15, 10, 18}$  u.s      changing 10 with 12

$\underline{8, 9, 10, 12, 15, 18}$  u.s

$8, 9, 10, 12, 15, 18 \rightarrow$  sorted.

9, 12, 15, 10, 8, 18, 20

9, 12, 10, 15, 8, 18, 20

9, 12, 10, 8, 15, 18, 20

9, 12, 10, 8, 15, 18, 20

(9, 12, 10, 8, 15, 18, 20)

#### 4<sup>th</sup> iteration

9, 12, 10, 8, 15, 18, 20

9, 12, 10, 8, 15, 18, 20

9, 10, 12, 8, 15, 18, 20

9, 10, 8, 12, 15, 18, 20

9, 10, 8, 12, 15, 18, 20

9, 10, 8, 12, 15, 18, 20

9, 10, 8, 12, 15, 18, 20

9, 10, 8, 12, 15, 18, 20

#### 5<sup>th</sup> iteration

9, 10, 8, 12, 15, 18, 20

9, 10, 8, 12, 15, 18, 20

9, 8, 10, 12, 15, 18, 20

#### 6<sup>th</sup> iteration

9, 8, 10, 12, 15, 18, 20

8, 9, 10, 12, 15, 18, 20

Ques:-

(18, 12, 9, 20, 15, 10, 8)

Bubble sorting

1<sup>st</sup> iteration

[18, 12, 9, 20, 15, 10, 8]

12, 18, 9, 20, 15, 10, 8

12, 9, 18, 20, 15, 10, 8

12, 9, 18, 20, 15, 10, 8

12, 9, 18, 15, 20, 10, 8

12, 9, 18, 15, 10, 20, 8

(12, 9, 18, 15, 10, 8, 20)

2<sup>nd</sup> iteration

12, 9, 18, 15, 10, 8, 20

9, 12, 18, 15, 10, 8, 20

9, 12, 18, 15, 10, 8, 20

9, 12, 15, 18, 10, 8, 20

9, 12, 15, 10, 18, 8, 20

9, 12, 15, 10, 8, 18, 20

(9, 12, 15, 10, 8, 18, 20)

3<sup>rd</sup> iteration

9, 12, 15, 10, 8, 18, 20

9, 12, 15, 10, 8, 18, 20

Day: \_\_\_\_\_

## Ques :-

Date: \_\_\_\_\_

(35, 38, 32, 39, 31, 33, 36, 40)

### Insertion sort

35, 38, 32, 39, 31, 33, 36, 40 u.s

35, 38, 32, 39, 31, 33, 36, 40 u.s

32, 35, 38, 39, 31, 33, 36, 40 u.s

32, 35, 38, 39, 31, 33, 36, 40 u.s

31, 32, 33, 35, 38, 39, 36, 40 u.s

31, 32, 33, 35, 36, 38, 39, 40 u.s

### Selection sorting

35, 38, 32, 39, 31, 33, 36, 40 u.s

31, 38, 32, 39, 35, 33, 36, 40 u.s

31, 32, 38, 39, 35, 33, 36, 40

31, 32, 33, 39, 35, 38, 36, 40

31, 32, 33, 35, 36, 38, 39, 40

31, 32, 33, 35, 36, 38, 39, 40

31, 32, 33, 35, 36, 38, 39, 40

28, 35, 9, 46, 18, 42, 50, 65, 76

28, 35, 9, 18, 46, 42, 50, 65, 76

28, 35, 9, 18, 42, 46, 50, 65, 76

### 5<sup>th</sup> iteration

28, 35, 9, 18, 42, 46, 50, 65, 76

28, 35, 9, 18, 42, 46, 50, 65, 76

28, 9, 35, 18, 42, 46, 50, 65, 76

28, 9, 18, 35, 42, 46, 50, 65, 76

28, 9, 18, 35, 42, 46, 50, 65, 76

### 6<sup>th</sup> iteration

28, 9, 18, 35, 42, 46, 50, 65, 76

9, 28, 18, 35, 42, 46, 50, 65, 76

9, 18, 28, 35, 42, 46, 50, 65, 76

9, 18, 28, 35, 42, 46, 50, 65, 76

### 7<sup>th</sup> iteration

9, 18, 28, 35, 42, 46, 50, 65, 76

9, 18, 28, 35, 42, 46, 50, 65, 76

9, 18, 28, 35, 42, 46, 50, 65, 76

### 8<sup>th</sup> iteration

9, 18, 28, 35, 42, 46, 50, 65, 76

9, 18, 28, 35, 42, 46, 50, 65, 76

46, 35, 28, 50, 65, 76, 9, 18, 42

46, 35, 28, 50, 65, 76, 9, 18, 42

46, 35, 28, 50, 65, 76, 9, 18, 42

46, 35, 28, 50, 65, 9, 18, 76, 42

46, 35, 28, 50, 65, 9, 18, 42, 76

## 2<sup>nd</sup> iteration

46, 35, 28, 50, 65, 9, 18, 42, 76

35, 46, 28, 50, 65, 9, 18, 42, 76

35, 28, 46, 50, 65, 9, 18, 42, 76

35, 28, 46, 50, 65, 9, 18, 42, 76

35, 28, 46, 50, 9, 18, 65, 42, 76

35, 28, 46, 50, 9, 18, 42, 65, 76

35, 28, 46, 50, 9, 18, 42, 65, 76

## 3<sup>rd</sup> iteration

35, 28, 46, 50, 9, 18, 42, 65, 76

28, 35, 46, 50, 9, 18, 42, 65, 76

28, 35, 46, 50, 9, 18, 42, 65, 76

28, 35, 46, 50, 9, 18, 42, 65, 76

28, 35, 46, 9, 50, 18, 42, 65, 76

28, 35, 46, 9, 18, 50, 42, 65, 76

28, 35, 46, 9, 18, 42, 50, 65, 76

## 4<sup>th</sup> iteration

28, 35, 46, 9, 18, 42, 50, 65, 76

28, 35, 46, 9, 18, 42, 50, 65, 76

28, 35, 46, 9, 18, 42, 50, 65, 76

ay: \_\_\_\_\_

## Selection sorting

102, 108, 107, 109, 117, 103, 120, 138, 128, 120 u.s

101, 108, 102, 109, 117, 103, 120, 138, 128, 120 u.s

101, 102, 103, 109, 117, 103, 120, 138, 128, 120 u.s

101, 102, 103, 109, 117, 108, 120, 138, 128, 120 u.s

101, 103, 103, 108, 117, 109, 120, 138, 138, 120 u.s

101, 103, 103, 108, 117, 109, 120, 138, 128, 120 u.s

101, 103, 103, 108, 109, 117, 120, 138, 128, 120 u.s

101, 102, 103, 108, 109, 117, 120, 138, 128, 120 u.s

101, 102, 103, 108, 109, 117, 120, 120, 128, 138

101, 102, 103, 108, 109, 117, 120, 120, 128, 138 u.s

(50, 46, 35, 28, 65, 76, 9, 18, 42) → sorted

## Selection sort

50, 46, 35, 28, 65, 76, 9, 18, 42 u.s

9, 46, 35, 28, 65, 76, 50, 18, 42 u.s

9, 18, 35, 28, 65, 76, 50, 46, 42 u.s

9, 18, 28, 35, 65, 76, 50, 46, 42 u.s

9, 18, 28, 35, 65, 76, 50, 46, 42 u.s

9, 18, 28, 35, 42, 65, 76, 50 u.s

9, 18, 28, 35, 42, 46, 50, 65, 76 → sorted

## Bubble sorting

### 1<sup>st</sup> iteration

50, 46, 35, 28, 65, 76, 9, 18, 42

46, 50, 35, 28, 65, 76, 9, 18, 42

46, 35, 50, 28, 65, 76, 9, 18, 42

5<sup>th</sup> iteration

101, 102, 103, 108, 109, 117, 120, 120, 128, 138

101, 102, 103, 108, 109, 117, 120, 120, 128, 138

101, 102, 103, 108, 109, 117, 120, 120, 128, 138

101, 102, 103, 108, 109, 117, 120, 120, 128, 138

101, 102, 103, 108, 109, 117, 120, 120, 128, 138

101, 102, 103, 108, 109, 117, 120, 120, 128, 138

6<sup>th</sup> iteration

101, 102, 103, 108, 109, 117, 120, 120, 128, 138

101, 102, 103, 108, 109, 117, 120, 120, 128, 138

101, 102, 103, 108, 109, 117, 120, 120, 128, 138

101, 102, 103, 108, 109, 117, 120, 120, 128, 138

101, 102, 103, 108, 109, 117, 120, 120, 128, 138

7<sup>th</sup> iteration

101, 102, 103, 108, 109, 117, 120, 120, 128, 138

101, 102, 103, 108, 109, 117, 120, 120, 128, 138

101, 102, 103, 108, 109, 117, 120, 120, 128, 138

101, 102, 103, 108, 109, 117, 120, 120, 128, 138

8<sup>th</sup> iteration

101, 102, 103, 108, 109, 117, 120, 120, 128, 138

101, 102, 103, 108, 109, 117, 120, 120, 128, 138

101, 102, 103, 108, 109, 117, 120, 120, 128, 138

9<sup>th</sup> iteration

101, 102, 103, 108, 109, 117, 120, 120, 128, 138

101, 102, 103, 108, 109, 117, 120, 120, 128, 138

w. 102, 101, 108, 109, 103, 117, 120, 128, 138. Date: \_\_\_\_\_

## 2<sup>nd</sup> iteration

102, 101, 108, 109, 103, 117, 120, 128, 120, 138

101, 102, 108, 109, 103, 117, 120, 128, 120, 138

101, 102, 108, 109, 103, 117, 120, 128, 120, 138

101, 102, 108, 109, 103, 117, 120, 128, 120, 138

101, 102, 108, 103, 109, 117, 120, 128, 120, 132

101, 102, 108, 103, 109, 117, 120, 128, 120, 138

~~101, 102, 108, 103, 109, 117, 120, 120, 128, 138.~~

## 3<sup>rd</sup> iteration

101, 102, 108, 103, 109, 117, 120, 120, 128, 138

101, 102, 108, 103, 109, 117, 120, 120, 128, 138

101, 102, 103, 108, 109, 112, 120, 120, 128, 138

101, 102, 103, 108, 109, 117, 120, 129, 138, 139

101, 102, 103, 108, 109, 112, 130, 130, 138, 138

101, 102, 103, 108, 109, 117, 130, 130, 138, 138

101, 102, 103, 108, 109, 112, 120, 122, 125, 127

## 4<sup>th</sup> iteration

101, 102, 103, 108, 109, 107, 120, 120, 128, 138

101, 102, 103, 108, 109, 117, 120, 120, 128, 138

101, 102, 103, 108, 109, 117, 120, 120, 128, 138

101, 102, 103, 108, 109, 117, 120, 120, 128, 139

101, 102, 103, 108, 109, 117, 120, 120, 128, 138

101, 102, 103, 108, 109, 117, 120, 120, 120, 132

101 102 103 108 109 110

, 102, 103, 108, 109, 117, 120, 120, 120, 128

Day:

**Q. NO. 1**

Date:

**(11, 1, 5, 4, 3, 2)****Insertion sort**11, 1, 5, 4, 3, 21, 11, 5, 4, 3, 21, 5, 11, 4, 3, 21, 4, 5, 11, 3, 21, 3, 4, 5, 11, 21, 2, 3, 4, 5, 111, 2, 3, 4, 5, 11 → sorted**Bubble sorting****(1<sup>st</sup> iteration)**11, 1, 5, 4, 3, 21, 11, 5, 4, 3, 21, 5, 11, 4, 3, 21, 5, 4, 11, 3, 21, 5, 4, 3, 11, 2**Q. NO. 2****(102, 108, 101, 109, 117, 103, 120, 138, 128)****Bubble Sort****(1<sup>st</sup> iteration)**102, 108, 109, 101, 117, 103, 120, 138, 128, 120102, 108, 101, 109, 117, 103, 120, 138, 128, 120102, 101, 108, 109, 117, 103, 120, 138, 128, 120102, 101, 108, 109, 103, 117, 120, 138, 128, 120102, 101, 108, 109, 103, 117, 120, 128, 138, 120102, 101, 108, 109, 103, 117, 120, 128, 138, 120102, 101, 108, 109, 103, 117, 120, 128, 138, 120102, 101, 108, 109, 103, 117, 120, 128, 138, 120

~~X~~ Q = 50, 46, 35, 28, 65, 76, 9, 18, 42

Pivot : 42

50 > 42

46 > 42

35 < 42

28 < 42

65) X J X

Question :-

50, 46, 35, 28, 65, 76, 9, 18, 42

0 : None

2 : 42

6 : 46, 76

9 : 9

1 : None

5 : 35, 50, 65

8 : 28, 18

Combine and separate by tens digit

0 : 9, 18, 28

3 : 35

4 : 42, 46

5 : 50

6 : 65, 76

Final sorted list :-

9, 18, 28, 35, 42, 46, 50, 65, 76

CSite S2345

- The place variable represents: the current place value (1 for ones to for tens.)

### Question

[102, 108, 101, 109, 117, 103, 120, 138, 128, 120]

### Radix sorting

Separate numbers by unit digits

0 : None                            1 : 101

2 : 102, 120, 120                3 : 103

8 : 108, 128                      9 : 109

7 : 117                            8 : 138

Compare and separate by tens digit

1 : 101, 102, 103, 108, 109

1 : 117

1 : 1020, 120

1 : 128

1 : 138

Final sorted list is :-

101, 102, 103, 108, 109, 117, 120, 120, 128, 138

## Quick Sorting

- Start the process.
- Choose first or last element as pivot.
- Arrange all the elements such that, the smaller number of pivot are on left side and greater are on the right side.
- Swap the pivot with element at position i as 1, placing in it's correct place.
- The function then returns the pivot index to Quick sort so array can be recursively divided into smaller parts.

## Radix Sorting

- Start the process
- It firstly finds the maximum number in the array to determine the number of digits we need to process.
- It then, iterates over the digits of the number starting from the last significant digits and uses counting sort the array based on each digit.

## Bubble sorting

- Compare adjacent elements in the array
- If the left element is greater than the right element, swap them
- Repeat this process for each element in array with each pass moving the largest **unsorted element** to its correct position

Continue until the array is sorted.

## Merge sorting

- Start the process
- The array is divided into two halves until each half has one or zero elements
- Each half is sorted individually by recursively applying mergesort
- The two sorted halves are merged into **single sorted array** using the merge function.

# Pseudocodes

Q: 11, 1, 5, 4, 3, 2 (Insertion sorting)

- Start with the second element (index 1) and compare it to the element before it
- Shift all elements greater than the current element to the right
- Insert the key at the correct position
- Repeat for all elements into the array.

(Selection sorting)

- Start with the first element and assume it as the smallest one.
- Find the smallest element in the unsorted part of the array.
- Swap the smallest element with the current element
- Move to next position and repeat it until the sorted part arrives.