

Group Presentation Assignment#2 and Guidelines

Assignment

1. Students will work together in groups of 2 to write a Python program on one of the sorting algorithms. For this occasion, students are not allowed to work in the same group as before the mid-term.
2. The running examples of an input for sorting should be as follows
 - a. {72, 90, 73, 53, 26, 61, 48, 32, 88}.
 - b. {C, Z, Y, A, X, D, U}.
3. The examples of expected output for 2a are given below for each sorting algorithm.
4. The presentation is expected to be on 20 October 2020.
5. The duration of presentation should be no longer than 10 minutes.
6. The presentation should focus on explaining the procedures behind the codes (i.e. how it works) and running examples.
7. It is not necessary to prepare a presentation slide.
8. However, the Python code in .ipynb or .py should be submitted before 11.59PM of 19 October 2020.
9. The output for both 2a and 2b should also be submitted with the Python code.

A) Bubble Sort

Expected input: data = [72, 90, 73, 53, 26, 61, 48, 32, 88]

Function name and argument: bubbleSort(data)

Expected output:

Original list: [72, 90, 73, 53, 26, 61, 48, 32, 88]

Round: 1

No exchange between 88 and 32

Exchange between 48 and 32

Exchange between 61 and 32

No exchange between 32 and 26

Exchange between 53 and 26

Exchange between 73 and 26

Exchange between 90 and 26

Exchange between 72 and 26

Current List: [26, 72, 90, 73, 53, 32, 61, 48, 88]

Round: 2

No exchange between 88 and 48

Exchange between 61 and 48

No exchange between 48 and 32

Exchange between 53 and 32

Exchange between 73 and 32

Exchange between 90 and 32

Exchange between 72 and 32

Current List: [26, 32, 72, 90, 73, 53, 48, 61, 88]

Round: 3

No exchange between 88 and 61

No exchange between 61 and 48

Exchange between 53 and 48

Exchange between 73 and 48

Exchange between 90 and 48

Exchange between 72 and 48

Current List: [26, 32, 48, 72, 90, 73, 53, 61, 88]

Round: 4

No exchange between 88 and 61

No exchange between 61 and 53

Exchange between 73 and 53

Exchange between 90 and 53

Exchange between 72 and 53

Current List: [26, 32, 48, 53, 72, 90, 73, 61, 88]

Round: 5

No exchange between 88 and 61

Exchange between 73 and 61

Exchange between 90 and 61

Exchange between 72 and 61

Current List: [26, 32, 48, 53, 61, 72, 90, 73, 88]

Round: 6

No exchange between 88 and 73

Exchange between 90 and 73

No exchange between 73 and 72

Current List: [26, 32, 48, 53, 61, 72, 73, 90, 88]

Round: 7

Exchange between 90 and 88

No exchange between 88 and 73

Current List: [26, 32, 48, 53, 61, 72, 73, 88, 90]

Round: 8

No exchange between 90 and 88

Current List: [26, 32, 48, 53, 61, 72, 73, 88, 90]

Final output: [26, 32, 48, 53, 61, 72, 73, 88, 90]

B) Selection Sort

Expected input: data = [72, 90, 73, 53, 26, 61, 48, 32, 88]

Function name and argument: selectionSort(data)

Expected output:

Original list: [72, 90, 73, 53, 26, 61, 48, 32, 88]

Round: 1

Position to insert: 0

Position of min value: 4

Current List: [26, 90, 73, 53, 72, 61, 48, 32, 88]

Round: 2

Position to insert: 1

Position of min value: 7

Current List: [26, 32, 73, 53, 72, 61, 48, 90, 88]

Round: 3

Position to insert: 2

Position of min value: 6

Current List: [26, 32, 48, 53, 72, 61, 73, 90, 88]

Round: 4

Position to insert: 3

Position of min value: 3

Current List: [26, 32, 48, 53, 72, 61, 73, 90, 88]

Round: 5

Position to insert: 4

Position of min value: 5

Current List: [26, 32, 48, 53, 61, 72, 73, 90, 88]

Round: 6

Position to insert: 5

Position of min value: 5

Current List: [26, 32, 48, 53, 61, 72, 73, 90, 88]

Round: 7

Position to insert: 6

Position of min value: 6

Current List: [26, 32, 48, 53, 61, 72, 73, 90, 88]

Round: 8

Position to insert: 7

Position of min value: 8

Current List: [26, 32, 48, 53, 61, 72, 73, 88, 90]

Final output: [26, 32, 48, 53, 61, 72, 73, 88, 90]

C) Insertion Sort

Expected input: data = [72, 90, 73, 53, 26, 61, 48, 32, 88]

Function name and argument: insertionSort(data)

Expected output:

Original list: [72, 90, 73, 53, 26, 61, 48, 32, 88]

Round: 1

Current Value: 90

Position of the current value: 1

Position to insert: 1

Current List: [72, 90, 73, 53, 26, 61, 48, 32, 88]

Round: 2

Current Value: 73

Position of the current value: 2

Position to insert: 1

Current List: [72, 73, 90, 53, 26, 61, 48, 32, 88]

Round: 3

Current Value: 53

Position of the current value: 3

Position to insert: 0

Current List: [53, 72, 73, 90, 26, 61, 48, 32, 88]

Round: 4

Current Value: 26

Position of the current value: 4

Position to insert: 0

Current List: [26, 53, 72, 73, 90, 61, 48, 32, 88]

Round: 5

Current Value: 61

Position of the current value: 5

Position to insert: 2

Current List: [26, 53, 61, 72, 73, 90, 48, 32, 88]

Round: 6

Current Value: 48

Position of the current value: 6

Position to insert: 1

Current List: [26, 48, 53, 61, 72, 73, 90, 32, 88]

Round: 7

Current Value: 32

Position of the current value: 7

Position to insert: 1

Current List: [26, 32, 48, 53, 61, 72, 73, 90, 88]

Round: 8

Current Value: 88

Position of the current value: 8

Position to insert: 7

Current List: [26, 32, 48, 53, 61, 72, 73, 88, 90]

Final output: [26, 32, 48, 53, 61, 72, 73, 88, 90]

D) Merge Sort

Expected input: data = [72, 90, 73, 53, 26, 61, 48, 32, 88]

Function name and argument: mergeSort(data)

Expected output:

Splitting [72, 90, 73, 53, 26, 61, 48, 32, 88]

Splitting [72, 90, 73, 53]

Splitting [72, 90]

Splitting [72]

Merging [72]

Splitting [90]

Merging [90]

Merging [72, 90]

Splitting [73, 53]

Splitting [73]

Merging [73]

Splitting [53]

Merging [53]

Merging [53, 73]

Merging [53, 72, 73, 90]

Splitting [26, 61, 48, 32, 88]

Splitting [26, 61]

Splitting [26]

Merging [26]

Splitting [61]

Merging [61]

Merging [26, 61]

Splitting [48, 32, 88]

Splitting [48]

Merging [48]

Splitting [32, 88]

Splitting [32]

Merging [32]

Splitting [88]

Merging [88]

Merging [32, 88]

Merging [32, 48, 88]

Merging [26, 32, 48, 61, 88]

Merging [26, 32, 48, 53, 61, 72, 73, 88, 90]

Final output: [26, 32, 48, 53, 61, 72, 73, 88, 90]

E) Quick Sort

Expected input: data = [72, 90, 73, 53, 26, 61, 48, 32, 88]

Function name and arguments: quickSort(data,0,len(data)-1)

Expected output:

Original List: [72, 90, 73, 53, 26, 61, 48, 32, 88]

Current Pivot (high): 88

Current low: 72

Exchange between 72 and 72

Current low: 90

Current low: 73

Exchange between 90 and 73

Current low: 53

Exchange between 90 and 53

Current low: 26

Exchange between 90 and 26

Current low: 61

Exchange between 90 and 61

Current low: 48

Exchange between 90 and 48

Current low: 32

Exchange between 90 and 32

Exchange between 90 and pivot 88

Partition/Split point = 7

Quicksort left sublist low = 0 high = 6

Original List: [72, 73, 53, 26, 61, 48, 32, 88, 90]

Current Pivot (high): 32

Current low: 72

Current low: 73

Current low: 53

Current low: 26

Exchange between 72 and 26

Current low: 61

Current low: 48

Exchange between 73 and pivot 32

Partition/Split point = 1
Quicksort left sublist low = 0 high = 0
Quicksort right sublist low = 2 high = 6
Original List: [26, 32, 53, 72, 61, 48, 73, 88, 90]
Current Pivot (high): 73
Current low: 53
Exchange between 53 and 53
Current low: 72
Exchange between 72 and 72
Current low: 61
Exchange between 61 and 61
Current low: 48
Exchange between 48 and 48
Exchange between 73 and pivot 73

Partition/Split point = 6
Quicksort left sublist low = 2 high = 5
Original List: [26, 32, 53, 72, 61, 48, 73, 88, 90]
Current Pivot (high): 48
Current low: 53
Current low: 72
Current low: 61
Exchange between 53 and pivot 48

Partition/Split point = 2
Quicksort left sublist low = 2 high = 1
Quicksort right sublist low = 3 high = 5
Original List: [26, 32, 48, 72, 61, 53, 73, 88, 90]
Current Pivot (high): 53
Current low: 72
Current low: 61
Exchange between 72 and pivot 53

Partition/Split point = 3
Quicksort left sublist low = 3 high = 2
Quicksort right sublist low = 4 high = 5
Original List: [26, 32, 48, 53, 61, 72, 73, 88, 90]
Current Pivot (high): 72
Current low: 61
Exchange between 61 and 61

Exchange between 72 and pivot 72

Partition/Split point = 5

Quicksort left sublist low = 4 high = 4

Quicksort right sublist low = 6 high = 5

Quicksort right sublist low = 7 high = 6

Quicksort right sublist low = 8 high = 8

Final output: [26, 32, 48, 53, 61, 72, 73, 88, 90]

Guidelines

- Each team member should contribute equally - no free riders!
- Each member should speak for approximately 7 to 8 minutes.
- Try to avoid small fonts as they are often difficult to read.
- Think of this presentation as explaining to your friends, not to business people.

Grading - 10% for each presentation

Grading is based on the presentation as a whole.

Assessment criterias

1. Comprehension: 2.5%

- Students are able to explain the topic to the audience.
- The solution provided is correct or convincing.

2. Content: 2.5%

- The content is clear and concise.

3. Creativity and contribution: 2.5%

- All group members speak equally.
- Creativity is highly encouraged to increase engagement.

4. Presentation skills: 2.5%

- All audience members can hear the presentation clearly.
 - Maintaining eye contact, and rarely reading the notes.
 - Not exceeding the time limit.
-
- Shyness or stumbling do not negatively affect the grade.
 - Reading the notes throughout the presentation is allowed, however, this will affect the grade.