جامعة الحسين التقنية
Al Hussein Technical University

ASSIGNEMNT BRIEF

HTU Course No:	40201201	HTU Course Name: Data Structures & Algorithms
BTEC UNIT No:	T/618/7430	BTEC UNIT Name: Data Structures & Algorithms

Version: 3

جامعة الحسين التقنية
Al Hussein Technical University

Assignment Brief

Student Name / ID Number	
HTU Course Number and Title	**40201201 Data Structures & Algorithms**
BTEC Unit Number and Title	**T/618/7430 Data Structures & Algorithms**
Academic Year	Spring 2022/2023
Assignment Author	Eng. Hana Rasheed Dr. Abdullah Amareen Dr. Safaa Hreiz Dr. Heba Alawneh Eng. Ashraf Smadi Eng. Dania Saeed Eng. Malek Allouzi
Course Tutor	Eng. Hana Rasheed Dr. Abdullah Amareen Dr. Safaa Hreiz Dr. Heba Alawneh Eng. Ashraf Smadi Eng. Dania Saeed Eng. Malek Allouzi
Assignment Title	Interview Questions
Assignment Ref No	**Assignment 1**
Issue Date	07/05/2023
Formative Assessment dates	14/05/2023 to 01/06/2023
Submission Date	18/06/2023
IV Name & Date	Dr. Murad Yaghi 06/05/2023

Submission Format

The submission of this assignment consists of Three parts:

- An individual written report that has a solution for tasks that ends with the word (**Report**).
- A full working source codes for tasks that require code implementation, these tasks will end with the word (**Code**).
- **Discussion** with your instructor (and any other witness) about the submitted code.

You need to follow the following guidelines, failing to follow them may result an 'unclassified' grade.

Assignment Guidelines

- You are required to submit a well formatted **PDF** report that provides a complete answer for all required tasks.
- Full and clear answers for all required tasks, mention the task number and the subtask number before each answer**.**
- You should sign the **student declaration form** attached to this assignment brief (use electronic signature to sign it).
- You must submit a compressed file (**ZIP** file) that includes all **Source codes** for the tasks that require code implementation, each source code should be named by the task number and the extension should be .java (*TaskN.java*). The source file should be **fully commented**.

- Soft-copy submissions are only allowed, you are required to upload your submission files to the university's eLearning platform through (https://elearning.htu.edu.jo/) within the submission date and time stated above. **NO SUBMISSION by EMAIL and NO LATE SUBMISSIONS WILL BE ACCEPTED.**
- If you commit any kind of plagiarism, HTU policies and regulations will be applied.
- The Discussion will be an online one to one oral discussion between you and your instructor through Microsoft Teams, which includes debugging, analyzing, and evaluating the code and algorithm developed in this code.
- The attendance of the oral discussion is mandatory in the date and time determined by your instructor, the exact discussion schedule will be announced after your submission, and you need to be ready to open your camera from the beginning of the discussion.
- You must sign the **witness form** that your instructor will fill up after the discussion to complete the discussion process.
- If you referred to any external resource for your answers, you need to mention this resource as a reference in **Reference**s section in your report.

جامعة الحسين التقنية
Al Hussein Technical University

Unit Learning Outcomes

LO1. Examine abstract data types, concrete data structures and algorithms.
LO2. Specify abstract data types and algorithms in a formal notation.
LO3. Implement complex data structures and algorithms.
LO4. Assess the effectiveness of data structures and algorithms.

Assignment Brief and Guidance

Scenario

You have applied to get an internship in a world class company in computer science filed, as a part of the interview process, you were given the following questions to get this internship, you need to answer these take-home questions clearly and correctly as a part of getting this opportunity:

Part 1:

Q1) Create a design specification **using linked list** for a Min-Priority Queue as an Abstract Data Type (ADT). Explain its formal definition, an application of it, and the three main operations including **insert**, **getMin**, and **removeMin**. Based on your design, you have to specify time complexity (Big-O) of each operation. **(Report)**

A min priority queue is a data structure that allows efficient insertion of elements and retrieval of the smallest element. It is a collection of items where each item has a priority associated with it, and the item with the highest priority is always at the front of the queue.
The primary operations on a min priority queue are insertion and deletion of elements. When an element is inserted into the queue, it is placed in a position based on its priority relative to the other elements in the queue. The element with the highest priority (i.e. minimum value) is placed at the front of the queue, while the element with the lowest priority (i.e. maximum value) is placed at the rear.
When an element is deleted from the queue, the element with the highest priority (i.e. minimum value, the one at the front of the queue) is removed.

Here is an example of Min-Priority Queue:

front

2	Ali
priority	data

5	Ahmad
priority	data

8	Hani
priority	data

rear

20	Jamal
priority	data

If we insert new element with priority = 6 and data = Sali then the priority queue will be:

front rear

2	Ali	5	Ahmad	6	Sali	8	Hani	20	Jamal
priority	data	priority	data	priority	data	priority	data	priority	data

Q2) Draw an illustration that simulates your design for the Min-Priority Queue. Start with an empty priority queue, then draw the priority queue after applying multiple operations. Keep in mind to apply all the valid operations that you have mentioned in the previous question. **(Report)**

Q3) Implement the Min-Priority Queue based on your proposed design, include the main three operations: **(Code)**

- *void insert(int priority, String data);*
- *String getMin();*
- *void removeMin();*

Part 2:

Examine the relation between the time complexity (Big-O) and the actual running time of Two sorting algorithms: Selection Sort and Merge Sort. Run both algorithms with two arrays:

- An already sorted array.
- A reversely sorted array.

Implement the following: **(Code)**

- **Merge sort** (the one discussed in class).
- **Selection Sort** (the one discussed in class).

Run each of the Two algorithms on the following array sizes: 5000, 50000, and 500000. For each program run, record the time the algorithm takes on the input size for each of the sorted and reversely sorted. Then, fill the following tables:

		5000	**50000**	**500000**
Selection Sort	Sorted			
	Reversely Sorted			

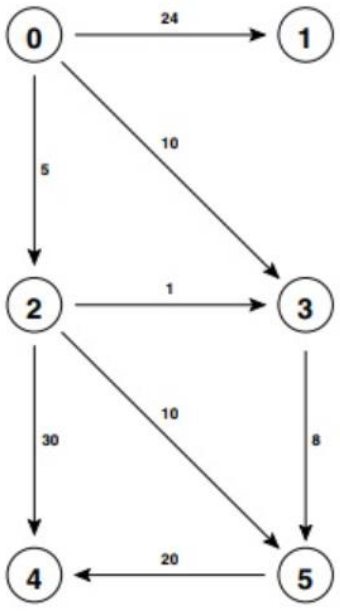
		5000	**50000**	**500000**
Merge Sort	Sorted			
	Reversely Sorted			

Answer each of the following questions based on the results in the tables above: **(Report)**

Q4) What is the theoretical time complexity (Big-O) of each of the sorting algorithms? Cover the worst case and best case if exists.

	Selection Sort	**Merge Sort**
Best case		
Worst case		

Q5) Compare the performance of the two sorting algorithms on ***sorted data*** and ***reversely sorted*** and explain which one is better and why.
Q6) How do you explain the results with the theoretical time complexity for each algorithm?
Q7) In the previous questions, you have used the running time measure to compare the efficiency between the Two algorithms. Briefly describe another measure that can be used to compare the efficiency of the algorithms.
Q8) Analyze the operations of the Dijkstra's algorithm and the Bellman-Ford algorithm by simulating both algorithms on the below graph starting from vertex 0 and ending in vertex 5. Your simulation should illustrate all the steps for each algorithm from the start of the algorithm until the end of the algorithm. Clearly state the shortest path extracted from each algorithm **(Report)**.



Q9) Implement the Dijkstra's Algorithm **(Code)**.
Q10) Your implementation should handle any expected errors that may occur when running the program. **(Code)**.

Part3:

A palindrome is a word, phrase, number, or other sequence of characters that reads the same forward and backward. In other words, it is a sequence of characters that is symmetrical when read in reverse order.

For example, the word "civic" is a palindrome because it reads the same forwards and backwards. Similarly, the number 1221 is a palindrome because it reads the same when its digits are reversed.

The following are examples for palindrome and non-palindrome sequences:

Sequence	Is it Palindrome?
Level	YES
Radar	YES
Madam	YES
Noon	YES
14341	YES
7557	YES
Car	NO
Ali	NO
1234	NO

Q11) Based on your understanding of the palindrome, write a Pseudocode for a function that takes a string and returns whether this string is palindrome or not. You must use a specific data structure to solve this problem.

Q12) Specify the data structure that you used in the previous question as abstract data type (ADT). Clearly specify its formal definition and its valid operations and illustrate these operations in a scenario from your choice using the logical representation of this data structure.

Q13) Show how the call stack can help in running the following code. Your illustrations should state the stack after each function call.

```java
public static void function2(){
    System.out.println("lets start ^_^");
}
public static void function1(int arr[],int i){
    if(i<0){
        function2();
    }
    else{
        function1(arr,i-1);
        System.out.println(arr[i]);
    }
}
public static void main(String[] args) {
   int arr[] = {1,2,3,4,5};
   function1(arr,4);
   function2();
}
```

Part 4:

Fill the following table with the time complexity of the operations based on the implemented data structure:

	Linked List	Array unsorted	Array sorted
Search			
Insert			
Remove			

Q14) Based on your analysis in the table above, choose the appropriate data structures for each scenario below **and justify your choice**: **(Report)**

1. Spell checker application. The data structure is used to store a dictionary of words. The data structure is constructed so that each node contains a word. When a user types a word, the spell checker searches for that word. If the word is found, it is considered spelled correctly. If the word is not found, the spell checker can suggest a list of words that are similar in spelling to the word that was typed, by searching for words that are similar in their prefix.

2. Priority queues. The data structure can be used to implement priority queues efficiently. It should allow for efficient insertion and deletion operations.

Q15) Evaluate the running time (Big-O) of the following code. Please elaborate on how you calculated the running time in detail. **(Report)**

```
public static int fun1(int [] arr, int index){
    if(index<=0){
        return arr[0];
    }
    int m1 = fun1(arr,index-1);
    int m2 = fun1(arr,index-2);
    int m3 = fun1(arr,index-4);
    if(m1>m2){
        return m1;
    }else if(m2>m3){
        return m3;
    }else{
        return m1;
    }
}
```

Q16) Examine the advantages of encapsulation and information hiding when implementing the stack or any other data structure as ADT.

Q17) Discuss the view that **imperative ADTs are a basis for object orientation** offering a justification for the view, and state if you agree.

Q18) As you know, the array and the linked list are implementation dependent, but the stack and queue are implementation independent. Evaluate three benefits of using implementation independent data structures.

Learning Outcomes and Assessment Criteria

Pass	Merit	Distinction
LO1 Examine abstract data types, concrete data structures and algorithms		**D1** Analyse the operation, using illustrations, of two network shortest path algorithms, providing an example of each.
P1 Create a design specification for data structures explaining the valid operations that can be carried out on the structures. **P2** Determine the operations of a memory stack and how it is used to implement function calls in a computer.	**M1** Illustrate, with an example, a concrete data structure for a First In First out (FIFO) queue. **M2** Compare the performance of two sorting algorithms.	
LO2 Specify abstract data types and algorithms in a formal notation		**D2** Discuss the view that imperative ADTs are a basis for object orientation and, with justification, state whether you agree.
P3 Using an imperative definition, specify the abstract data type for a software stack.	**M3** Examine the advantages of encapsulation and information hiding when using an ADT.	
LO3 Implement complex data structures and algorithms		**D3** Critically evaluate the complexity of an implemented ADT/algorithm.
P4 Implement a complex ADT and algorithm in an executable programming language to solve a well-defined problem. **P5** Implement error handling and report test results.	**M4** Demonstrate how the implementation of an ADT/algorithm solves a well-defined problem.	
LO4 Assess the effectiveness of data structures and algorithms		

P6 Discuss how asymptotic analysis can be used to assess the effectiveness of an algorithm. **P7** Determine two ways in which the efficiency of an algorithm can be measured, illustrating your answer with an example.	**M5** Interpret what a trade-off is when specifying an ADT using an example to support your answer.	**D4** Evaluate three benefits of using implementation independent data structures.

جامعة الحسين التقنية
Al Hussein Technical University

STUDENT ASSESSMENT SUBMISSION AND DECLARATION

When submitting evidence for assessment, each student must sign a declaration confirming that the work is their own.

Student name: **Student ID:**		**Assessor name:**
Issue date: / /	**Submission date:** / /	**Submitted on:** / /
Programme:		
HTU Course Name: Data Structures & Algorithms **BTEC UNIT Title**: Data Structures & Algorithms **HTU Course Code:** 40201201 **BTEC UNIT Code:** T/618/7430 **I AM REPEATING THIS UNIT*:** **(YES)** **(NO)**		

Plagiarism

Plagiarism is a particular form of cheating. Plagiarism must be avoided at all costs and students who break the rules, however innocently, may be penalised. It is your responsibility to ensure that you understand **correct referencing practices**. As a university level student, you are expected to use appropriate references throughout and keep carefully detailed notes of all your sources of materials for material you have used in your work, including any material downloaded from the Internet. Please consult the relevant unit lecturer or your course tutor if you need any further advice.

Student declaration

I certify that the assignment submission is entirely my own work and I fully understand the consequences of plagiarism. I understand that making a false declaration is a form of malpractice.

Student signature: **Date:**