

**Data Structures & Algorithms (DSA)**

Year 2/3 (2020/21), Semester 4/6

## SCHOOL OF INFOCOMM TECHNOLOGY

Diploma in Cybersecurity & Digital Forensics

Diploma in Information Technology

**TEST 1 – SOLUTION DOCUMENT**

INSTRUCTIONS TO CANDIDATES:

1. Write your Student Number, Name and Module Group CLEARLY in the boxes provided below.
2. Provide your answers to the questions in the Test 1 paper in this document.
3. Save this file as "Test1 – s1234567 Solution.docx" where s1234567 is your student number.
4. Map to network drive: [**\\ictspace.ict.np.edu.sg\DSATest1\**](file:///\\ictspace.ict.np.edu.sg\DSATest1\)
5. Copy this solution file into the network drive.

**ictspace.ict.np.edu.sg > DSATest1 > group > studentID**

|  |  |
| --- | --- |
| **Student Number: 10198239H** | **Seat Number:** |
| **Student Name: Pan YeCheng** | **Module Group: P06** |

**GRADE**

There are 3 questions. Answer ALL questions (100 marks).

Write your solutions to the questions in the space allocated for each question.

Question 1 – Solution (30 marks)

|  |  |
| --- | --- |
| (a) | void List::reverse()  {  List temp[MAX\_SIZE];  for (int i = getLength(); i < 0; i--) {  temp->add(items[i]);  }  for (int i = 0; i < MAX\_SIZE; i++) {  items[i] = temp->get(i);  }    } |
|  | (20 marks) |
| (b) | O(n2) – double for loop to move the item from back to start into a temp array and then bringing it back to the actual list. |
|  | (10 marks) |

Question 2 – Solution (40 marks)

|  |  |
| --- | --- |
| (a) | void List::reverse()  {  Node\* temp2 = firstNode;  Node\* temp = new Node;  int tempsize = getLength();  cout << tempsize;  for(int i= tempsize;i < 0;i--){  for (int i = 0; i < tempsize; i++) {  temp2->next;  }  temp->item= temp2->item;  temp->next;  }  for (int i = 0; i < size; i++) {  firstNode->item = temp->item;  firstNode->next;  temp->next;  }  } |
|  | (20 marks) |
| (b) | O(n2) – used double for loop to transfer it to another list |
|  | (5 marks) |
| (c) | void List::reverseR()  {  Node\* temp2 = firstNode;  Node\* temp = new Node;  int tempsize = getLength();  for (int i = tempsize; i < 0; i--) {  for (int i = 0; i < tempsize; i++) {  temp2->next;  }  temp->item = temp2->item;  temp->next;  }  reverseR(temp);    }  void List::reverseR(Node\*& head)  {  for (int i = 0; i < size; i++) {  firstNode->item = head->item;  firstNode->next;  head->next;  }  } |
|  | (15 marks) |

Question 3 – Solution (30 marks)

|  |  |
| --- | --- |
| (a) |  |
|  | (10 marks) |
| (b) |  |
|  | (10 marks) |
| (c) | O(n) – accessing the list through will use a time complexity of o(n). |
|  | (10 marks) |

**– End of Document –**