



Philadelphia University
Faculty of Engineering

Marking Scheme

Exam Paper

BSc CE

Algorithms and Data Structures (630231)

First Exam

First semester

Date: 28/11/2010

Section 1

Weighting 15% of the module total

Lecturer:

Dr. Qadri Hamarsheh

Coordinator:

Dr. Qadri Hamarsheh

Internal Examiner:

Dr. Ali Al-Khawaldeh

Marking Scheme

Algorithms and Data Structures (630231)

The presented exam questions are organized to overcome course material through 5 questions. The *all questions* are compulsory requested to be answered.

Marking Assignments

Question 1 This question is attributed with 3 marks if answered properly; the answers are as following:

1. In an array list, the time complexity of the `removeAt` function is identical to the time complexity of the ____ function.
 - a) `isEmpty`
 - b) `seqSearch`
 - c) `isFull`
 - d) `isItemAtEqual`
2. Building a linked list forward places the item to be added at the ____ of the linked list.
 - a) beginning
 - b) `end`
 - c) middle
 - d) key point
3. Because initially the list is empty, the pointer `first` must be initialized to _____.
 - a) `NULL`
 - b) `NIL`
 - c) `EMPTY`
 - d) `NOP`

Question 2 This question is attributed with 3 marks if answered properly; the answers are as following:

Question 2.a Solution:

$$f(n) = n^2 + 2n + 1 \text{ is } O(n^2)$$

For $n > 1$ we have

$$f(n) = n^2 + 2n + 1 \leq n^2 + 2n^2 + n^2 \leq 4n^2$$

$$f(n) \leq c \cdot g(n) \leq 4n^2$$

$$c = 4, g(n) = n^2, n = 1$$

(1.5 marks)

Question 2.b Solution:

$O(n^2 \log n)$: Outer loop is $O(n)$, inner loop is $O(n)$, but sorting is $O(n \log n)$. So, the complexity of the algorithm is $n(n + n \log n) = O(n^2 \log n)$ (1.5 marks)

Question 3 This question is attributed with 3 marks if answered properly. The complete code for this question as the following:

```
//in specification file complexType.h
complexType operator-(const complexType& otherComplex) const;           (1 mark)
//in implementation file complexType.cpp
complexType complexType::operator-(const complexType& otherComplex) const
{
    complexType temp;
    temp.realPart = realPart - otherComplex.GetRealPart();
    temp.imaginaryPart = imaginaryPart - otherComplex.GetImaginaryPartPart();
    return temp;
}
(2 marks)
```

Question 4 This question is attributed with 3 marks if answered properly. The complete code for this question as the following:

```
//in specification file arrayListType.h
void removeAll(const elemType& removeItem);
//in implementation file arrayListType.cpp
template <class elemType>
void arrayListType<elemType>::removeAll(const elemType& removeItem)
{
    int loc;
    if (length == 0)
        cerr << "Cannot delete from an empty list." << endl; (1.5 marks)
    else
    {
        loc = 0;
```

```

        while (loc < length)
            if (list[loc] == removeItem)
                removeAt(loc);
            else
                loc++;
    }
}

```

(1.5 marks)

Question 5 This question is attributed with 3 marks if answered properly. The answer is the following:

linkedListType
- copyList(const linkedListType<Type>& otherList): void
count: int
*first: nodeType<Type>
*last: nodeType<Type>
+ operator= (const linkedListType<Type>&): const linkedListType<Type>&
+ initializeList():void
+ isEmptyList() const: bool
+ print() const: void
+ length() const: int
+ destroyList():void
+ front() const: Type
+ back() const: Type
+ search(const Type& searchItem) const = 0: bool
+ insertFirst(const Type& newItem) = 0: void
+ insertLast(const Type& newItem) = 0: void
+ deleteNode(const Type& deleteItem) = 0: void
+ begin():LinkedListIterator<Type>
+ end():linkedListIterator<Type>
+linkedListType();
+linkedListType(const linkedListType<Type>& otherList);
+~linkedListType();