



King Saud University

College of Computer and Information Sciences

Department of Computer Science

Data Structures CSC 212**Midterm Exam 1 Sample Solution - Spring 2020**

Date: -

Duration: 90 minutes

Guidelines

- No calculators or any other electronic devices are allowed in this exam.

Student ID:

Name:

Section:

Instructor:

1	2	3.1	3.2	Total

Question 1 20 points

1. The list 1: A, B, C, D, E , after calling $1.f(3)$, 1 becomes:

- ☐ (A) B, C, D, E
☒ (B) D, A, B, C, E
☐ (C) E, B, C, D
☐ (D) A, D, E, B, C
☐ (E) None

2. The list 1: A, B, C, D, E , after calling $1.f(1)$, 1 becomes:

- ☐ (A) A, B, C
☐ (B) E, A, B, C, D
☐ (C) B, C, D, E, A
☒ (D) B, A, C, D, E
☐ (E) None

3. The list 1: A, B, C, D, E , after calling $1.f(5)$, 1 becomes:

- ☐ (A) A
☐ (B) E, A, B, C, D
☐ (C) A, B, C, D, E
☒ (D) E, A, B, C, D
☐ (E) None

4. The list 1: A, B, C, D, E , after calling $1.f(2)$, 1 becomes:

- ☐ (A) empty
 ☐ (B) E, D, C, B, A
☒ (C) C, A, B, D, E
☐ (D) E, C, D
☐ (E) None

Question 2 25 points

```

public static <T> void moveToEnd(List<T> l, int i)
{
    l.findFirst();
    for (int j = 0; j < i; j++)
        l.findNext();
    T e = l.retrieve();
    l.remove();
    while (!l.last())
        l.findNext();
    l.insert(e);
}

```

Question 3 55 points

1.

```

public static <T> PQ2Elem<T> minMax(PQ2<T> q) {
    Queue<PQ2Elem<T>> tq = new LinkedQueue<PQ2Elem<T>>();
    PQ2Elem<T> res = q.serve1();
    int maxPr1 = res.pr1;
    int minPr2 = res.pr2;
    q.enqueue(e);
    while (q.length() > 0) {
        PQ2Elem<T> e = q.serve1();
        tq.enqueue(e);
        if (e.pr1 == maxPr1 && e.pr2 < minPr2) {

```

```

        res = e;
        minPr2 = e.pr2;
    }
}
while(tq.length() > 0) {
    PQ2Elem<T> e = tq.serve();
    q.enqueue(e.data, e.pr1, e.pr2);
}
return res;
}

```

2. Here are two possible implementations:

- **Array implementation:**

```

public class ArrayPQ2<T> implements PQ2<T> {
    private int size, maxSize;
    private T[] data;
    private int[] pr1A;
    private int[] pr2A;
    public ArrayPQ2() {
        this.maxSize = maxSize;
        size = 0;
        data = (T[]) new Object[maxSize];
        pr1A = new int[maxSize];
        pr2A = new int[maxSize];
    }
    public int length() {
        return size;
    }
    public boolean full() {
        return size == maxSize;
    }
    public void enqueue(T e, int pr1, int pr2) {
        data[size] = e;
        pr1A[size] = pr1;
        pr2A[size] = pr2;
        size++;
    }
    private void remove(int k) {
        for (int i = k; i < size - 1; i++) {
            data[i] = data[i + 1];
            pr1A[i] = pr1A[i + 1];
            pr2A[i] = pr2A[i + 1];
        }
        size--;
    }
    public PQ2Elem<T> serve1() {
        int max = 0;
        for (int i = 1; i < size; i++)
            if (pr1A[i] > pr1A[max])
                max = i;
        PQ2Elem<T> e = new PQ2Elem<T>(data[max], pr1A[max], pr2A[max]);
        remove(max);
        return e;
    }
    public PQ2Elem<T> serve2() {
        int min = 0;
        for (int i = 1; i < size; i++)
            if (pr2A[i] < pr2A[min])
                min = i;
        PQ2Elem<T> e = new PQ2Elem<T>(data[min], pr1A[min], pr2A[min]);
        remove(min);
        return e;
    }
}

```

- **Linked implementation:**

```

class PQ2Node<T> {
    public T data;
    public int pr1;
    public int pr2;
    PQ2Node<T> next;
    public PQ2Elem(T data, int pr1, int pr2) {
        this.data = data;
        this.pr1 = pr1;
        this.pr2 = pr2;
        next = null;
    }
}

public class LinkedPQ2<T> implements PQ2<T> {
    private int size;
    private PQ2Node<T> head, tail;
    public LinkedPQ2() {
        head = tail = null;
        size = 0;
    }
    public int length() {
        return size;
    }
    public boolean full() {
        return false;
    }
    public void enqueue(T e, int pr1, int pr2) {
        PQ2Node<T> tmp = new PQ2Node<T>(e, pr1, pr2);
        if (tail == null)
            head = tail = tmp;
        else
            tail = tail.next = tmp;
        size++;
    }
    private void remove(PQ2Node<T> q, PQ2Node<T> p) {
        if (p == head) {
            head = head.next;
            if (head == null)
                tail = null;
        } else {
            q.next = p.next;
            if (p == tail)
                tail = q;
        }
        size--;
    }
    public PQ2Elem<T> serve1() {
        PQ2Node<T> max = head, pmax = null;
        PQ2Node<T> p = head, q = null;
        while (p != null) {
            if (p.pr1 > max.pr1) {
                max = p;
                pmax = q;
            }
            q = p;
            p = p.next;
        }
        remove(pmax, max);
        return new PQ2Elem<T>(max.data, max.pr1, max.pr2);
    }
    public PQ2Elem<T> serve2() {
        PQ2Node<T> min = head, pmin = null;
        PQ2Node<T> p = head, q = null;
        while (p != null) {
            if (p.pr2 < min.pr2) {
                min = p;
                pmin = q;
            }
            q = p;
            p = p.next;
        }
        remove(pmin, min);
        return new PQ2Elem<T>(min.data, min.pr1, min.pr2);
    }
}

```

```
        }
        q = p;
        p = p.next;
    }
    remove(pmin, min);
    return new PQ2Elem<T>(min.data, min.pr1, min.pr2);
}
}
```