

סמסטר ב', תשס"ז, 2006/7

בחינת מועד א'

**דף תשובות**

שאלה 1 סעיף א' (15 נקודות)

```
public static void sort (int [] arr, int left, int right) {  
    if (right > left) {  
        int mid = place (arr, left, right);  
        arrange(arr, left, mid, right);  
        sort(arr, left, mid - 1);  
        sort(arr, mid + 1, right);  
    }  
}
```

שאלה 1 סעיף ב' (10 נקודות)

```
private static int place (int [] arr, int left, int right) {  
    int mid = left;  
    for (int i = left+1; i <= right; ++i)  
        if (arr[i] < arr[left])  
            ++mid;  
    swap(arr, left, mid);  
    return mid;  
}
```

שאלה 1 סעיף ג' (10 נקודות)

```
private static void arrange (int [] arr, int left, int mid, int right) {  
    while (left < mid) {  
        while (left < mid && arr[left] < arr[mid])  
            ++left;  
        if (left < mid)  
            while (arr[right] > arr[mid])  
                --right;  
        swap(arr, left, right);  
    }  
}
```

## שאלה 2 (15 נקודות)

```
private static void subs (String s, int x) {
    out.println("\\"+s+"\");
    for (int i = x; i < s.length(); ++i)
        subs(s.substring(0, i)+s.substring(i+1), i);
}
```

## שאלה 3 סעיף א' (10 נקודות)

```
public Link whereIs (Object x) {
    if (next.data.equals(x))
        return this;
    Link p = next;
    while (p != this & !p.next.data.equals(x))
        p = p.next;
    if (p == this)
        return null;
    return p;
}
```

## שאלה 3 סעיף ב' (10 נקודות)

```
public void insertBefore (Object x, Link link) {
    Link p = whereIs(x);
    if (p != null) {
        Link q = link;
        while (q.next != link)
            q = q.next;
        q.next = p.next;
        p.next = link;
    }
}
```

## שאלה 4 סעיף א' (10 נקודות)

```
public class RootsIterator implements Iterator {

    private Function f;
    private int ub;
    private int current;

    public RootsIterator(Function f, int lb, int ub) {
        this.f = f;
        this.ub = ub;
        this.current = lb - 1;
        advance();
    }

    public boolean hasNext() {
        return current <= ub;
    }

    public int next() {
        int t = current;
        advance();
        return t;
    }
}
```

```

    }

    private void advance () {
        ++current;
        while (current <= ub) {
            if (f.mapOf(current) == 0)
                break;
            ++current;
        }
    }
}

```

#### שאלה 4 סעיף ב' (10 נקודות)

```

public abstract class Func implements Function, Iterable {

    public Iterator iterator(int lb, int ub) {
        return new RootsIterator(this, lb, ub);
    }

    public int max (int lb, int ub) {
        int m = mapOf(lb);
        for (int p = lb + 1; p <= ub; ++p) {
            int y = mapOf(p);
            if (m < y)
                m = y;
        }
        return m;
    }
}

```

#### שאלה 4 סעיף ג' (10 נקודות)

```

class CompositeFunc extends Func {

    private Function f1, f2;

    public CompositeFunc(Function f1, Function f2) {
        this.f1 = f1;
        this.f2 = f2;
    }

    public int mapOf(int x) {
        return f1.mapOf(f2.mapOf(x));
    }

    public boolean equals (Object x) {
        if (!(x instanceof CompositeFunc))
            return false;
        CompositeFunc other = (CompositeFunc) x;
        return f1.equals(other.f1) & f2.equals(other.f2);
    }
}

```