

CMPS 253: SOFTWARE ENGINEERING, Fall 2010 – 2011
Final Examination
Saturday January 29, 8:00 – 10:00 a.m., Nicely 415

Instructions.

- Please leave the first page blank: write only your name and section number.
- Please draw a horizontal line across the page between the answers to each question.
- You may refer to the following during the exam: the course textbook, the course lecture notes, your homework solutions, any notes that you have taken in class. If you wish to use existing results, please cite them properly, e.g., use a page number for results in the course textbook, or a lecture title and page number for results in the course notes.
- You may **not** refer to any other materials.

Good luck!

1. (40 points) Consider the following procedure:

```
public static int values(int[] a)
//REQUIRES: a is not null
//EFFECTS: returns the number of distinct values in a, i.e., returns
//    | { v : (exists i : 0 <= i < a.length : v = a[i]) } |
```

1a. (20 points) Provide an implementation (code) for `values` that satisfies the following conditions:

1. It contains at most one while loop
2. It calls a single helper procedure

1b. (10 points) Provide a specification for the helper, i.e., `REQUIRES` and `EFFECTS` clauses.

1c. (10 points) Provide an implementation for the helper. The implementation of the helper must contain at most one while loop.

Sample solution:

```
public static int values(int[] a) {
//REQUIRES: a is not null
//EFFECTS: returns the number of distinct values in a, i.e., returns
//    vals(a, a.length) where
//    vals(a, i) = | { v : (exists j : 0 <= j < i : v = a[j]) } |

int i = 0;    //loop counter
int[] b = new int[a.length];
int k=0;[      //index into b; b[0...k-1], which contains all
              //nonduplicated values in a[0...i-1]

//invariant: b[0...k-1] = all nonduplicated values in a[0...i-1]

while(i < a.length) {
    //b[0...k-1] = all nonduplicated values in a[0...i-1]
    if (!contains(a[i],b,k)) {
        //b[0...k-1] = all nonduplicated values in a[0...i-1] /\ a[i] notin b[0...k-1]
```

```

        b[k] = a[i];
        //b[0...k] = all nonduplicated values in a[0...i]
        k = k+1;
        //b[0...k-1] = all nonduplicated values in a[0...i]
    }
    else {
        //b[0...k-1] = all nonduplicated values in a[0...i-1] /\ a[i] in b[0...k-1]
        //b[0...k-1] = all nonduplicated values in a[0...i]
    }
    //b[0...k-1] = all nonduplicated values in a[0...i]
    i = i+1;
    //b[0...k-1] = all nonduplicated values in a[0...i-1]
}
//b[0...k-1] = all nonduplicated values in a[0...a.length-1]
return(k);
}

```

```

public static boolean contains(v, int[] b, k) {
    //REQUIRES: b is not null
    //EFFECTS: returns true iff v occurs in b[0...k-1]
    // otherwise returns false

    int i = 0;
    while(i < k) {
        if (v == b[i]) return(true);
        i = i+1;
    }
    return(false);
}

```

2. (40 points) This question concerns the implementation of an abstract data type: a sequence of integers, i.e., x_0, x_1, \dots, x_{n-1} where each x_i is an integer. Call the class `IntSeq`.

2a. (10 points) Determine an appropriate representation, i.e., instance variables for the class `IntSeq`.

Sample Solution. java arraylist or vector with start (bottom) and end (top) index vars:

```

arraylist[] a;
int st=0, end=0;

```

2b. (10 points) Give appropriate abstraction function and representation invariant.

Sample Solution. rep inv: $\text{end} \geq \text{st}$

$\text{AF}(a, \text{st}, \text{end}) = a[\text{st}], a[\text{st}+1], \dots, a[\text{end}-1]$ where $,$ denotes concatenation to form a sequence

2c. (20 points) Write a specification and implementation for the following operations, in a style similar to what I showed for `IntSet` :

- delete the leftmost element, e.g., so that deleting the leftmost element of x_0, x_1, \dots, x_{n-1} results in x_1, \dots, x_{n-1}
- insert an element v at the right end, e.g., so that inserting v into x_0, x_1, \dots, x_{n-1} results in $x_0, x_1, \dots, x_{n-1}, v$

`delete()`:

EFFECTS: `AF_post = tail(AF)`

`insert(v)`:

EFFECTS: `AF_post = AF ^ v` where `^` is concatenation

3. (20 points) Consider the file system specification discussed in class. We wish to model the size of files and directories. For a file, this is simply its size in bytes. For a directory d , this is the total size of all objects in the subtree with d as root.

3a. (5 points) Show the needed modifications to the file system specification data model, including any additional constraints, if needed.

3b. (15 points) Write a specification for an operation that returns the size of a file system object. In particular, give the formal versions of the **REQUIRES** and **EFFECTS** clauses.

Sample Solution.

3a. Add a `Size` data item which is a number (in bytes), and a relation `size` from `FSubject` to `size`. For file f include constraint that $f.size = |f.insides|/8$.

3b. `size(FSubject o)`

EFFECTS: if o is a file then return `o.size`

else return `(SUM o2: o2 in o.children: o2.size)`