

UNIVERSITY OF CALIFORNIA
Department of Electrical Engineering
and Computer Sciences
Computer Science Division

CS61B
Summer 2000

M. Brudno

Homework #3

Due: Tuesday, July 11, 2000

Create a directory to hold your answers to this homework set. You will find templates for some problems in `~/cs61b/hw/hw3`; please use them. To submit your homework, use the command `'submit hw3'` from within that directory. Where the problems specify file names to use for your solution, use exactly those names. Put your answers to non-programming questions in a file called `hw3.txt`.

1. Implement the following without using the operations `+`, `-`, `/`, `*`, or `%`.

```
class Quaternary {
    /** A string consisting of the integer N in base-4 notation
     * E.g., quaternary(5) = '11', quaternary(-12) = '-30'
     */
    static String quaternary(int x) { /* Fill in */ }
}
```

2. Implement the following without using the operations `+`, `-`, `/`, `*`, or `%`, or any other classes from the Java library or elsewhere.

```
class Adder {
    /** The value of x+y
     */
    static int add(int x, int y) { /* Fill in */ }
}
```

3. Prove that for any Java `int` value `x`, the expression `(~x)+1` yields `int -x`.
4. Assume that a digit from base `b` costs `b` dollars: a bit costs \$2, a base ten digit costs \$10. What is the most cost-efficient base? That is if you were given a set amount of money, by using which base would you be able to represent the largest numbers.
5. Given an array with positive integers, we wish to replace each integer with the closest to the right larger one, if no such element exists it should be replaced with 0. Given the array `{ 1, 5, 3, 7, 2, 3 }` we should get `{ 5, 7, 7, 0, 3, 0 }`. Implement this function so that it takes time linear to the length of the array.

```

class Shadow {
    /** Changes each number to the closest to the right larger one
     * in O(n) time.
     */
    static int[] shadow(int[] x) { /* Fill in */ }
}

```

6. In lecture we have seen queues. Now we will implement a priority queue datatype. Each element in the queue will have a priority. When `next()` is called on a queue, it should return the element in the queue which has the highest priority. You will have to implement two different priority queues: one using arrays, the other using linked lists. Both should inherit from the `abstract class Queue` from lecture notes. Please see the skeleton files for the details.

7. The equation $A^5 + B^5 + C^5 + D^5 + E^5 = F^5$ has only one solution that satisfies $0 < A \leq B \leq C \leq D \leq E \leq F \leq 75$. Write a program to find this solution.

8. Suppose that instead of using `null` to represent the end of a singly linked list, we use a list object whose `tail` is `null` and whose `head` is irrelevant (so that there is always at least one `List` object in the list). This special kind of list item is known as a *sentinel*. Fill in the following to match the comments.

```

class List2 {
    public Object head;
    public List2 tail;

    /** The empty List. */
    public List2() { tail = null; }

    /** Insert item X at the beginning of this list. That is, after
     * L.insert(X), L.tail points to a list containing the same items
     * that L originally pointed to and L.head is X. */
    public void insert(Object X) { /* FILL IN */ }

    /** Remove the first item in this list. Assumes !isEmpty(). That
     * is, after L.delete(), L points to a list containing the same
     * items that L.tail originally pointed to. */
    public void delete() { /* FILL IN */ }

    /** True iff there are no items in this list. */
    public boolean isEmpty() { return tail == null; }
}

```