Homework Assignment 1 - Iterative Algorithms and Framework

Prewords

- Deadline for submission is 09-27-2017 11:59 PM (PST).
- Please submit your assignment online at the submission page.
- Please upload a pdf file with the following naming conventions LastnameFirstName_HW1.pdf
- If you submit a hand-written scan of your homework, please submit also a physical copy of your homework through the homework box.
- This is NOT a group assignment!!! You are allowed to ask everybody and use every kind of help - BUT you need to submit your own solution.

Grading

We will grade your work according to the following criteria:

- Question 1 - 20 Points - 5 per prove
- Question 2 - 10 Points
- Question 3 - 20 Points
- Question 4 - 20 Points

Problem sets

Problem 1.
Prove the following assertions by using the definition of the asymptotic notations.

a) Show $4n + 8 = O(n)$

b) Show $15n^2 - 50 = \theta(n^2)$

c) Show n is not in $O(\sqrt{n})$

d) Show $n^2 = O(10^n)$

Problem 2.
Consider the following algorithm.

```
Secret1(n)
        {
        i=0; j=0; k=0;
                for (i = 0; i<n; i++)
                        {
                        for (j = 0; j<= n; j++ )
                                {
                                G();
                                G();
                                }
                        for (k = 0; k<n; k++)
                                G();
                        G();
                        G();
                        G();
                }
                G();
                G();
        }
```

Assuming the cost of the basic operation G() is 2 unit.

a) What is the concrete runningtime (T(n))?

b) What is the efficiency class of this algorithm? Prove that!

Problem 3.
Prove the correctness of algorithm ALG(A). Hint: algorithm return $2A$.

```
ALG(A)
{
        R = 0
        I = A
        while (I>0)
        {
                R = R + 2
                I = I - 1
        }
        return R
}
```

Problem 4.
Prove correctness of the following algorithm:

```
ALG(A)
{
        R = 0
        I = 2
        while (I<=A)
        {
                if ( I is even ) {  R = R + 1  }
                I = I + 1
        }
return R
}
```

Hint: algorithm returns $\frac{A}{2}$

Hint: Invariant (assuming indexing starts at 0):

$$\begin{cases} R_k = & \lfloor \frac{k+1}{2} \rfloor \\ I_k = & k+2 \end{cases}$$

An equivalent invariant (simpler, but longer proof – you will have to consider 2 cases each time):

$$\begin{cases} R_k = (I_k - 1)/2 \text{ if } I_k \text{ is odd} \\ R_k = (I_k - 2)/2 \text{ otherwise} \end{cases}$$