CS340 Analysis of Algorithms
Spring 2025

Worksheet:	**7**	**Professor:**	**Dianna Xu**
Date:		**E-mail:**	dxu@cs.brynmawr.edu
Title:	**Recurrences**	**URL:**	**http://cs.brynmawr.edu/cs340**

1 Recursive Binary Search

Consider the following pseudo code, which searches for a number x in an array $A[1,...,m]$ by calling `BSearch(A, x, 1, m)`

```
Function BSearch(array A, integer x, integer left, integer right)
    if left > right
     then
        return −1
    end
    m = (left+right)/2
    if A[m] = x
     then
        return m
    end
    else
        if A[m] > x
         then
            return BSearch(A, x, left, m-1)
        end
    end
    else
        return BSearch(A, x, m+1, right)
    end
```

Note that the function operates on an input size of `right - left + 1`. Write the recurrence $T(n)$ that denotes the maximum number of steps `BSeach` makes on an input of size n. In particular,

1. In the pseudocode above, highlight the non-recursive parts and estimate their running time in big-Oh notation.

2. In the pseudocode above, highlight the recursive parts and estimate their running time using $T(n)$.

3. Given the recurrence for $T(n)$. Don't worry about rounding. Do not forget the base case.

2 Master Theorem

Consider the following pseudo code segments, each of which has as input an array A of size n. State the recurrrence run time for each, using $T(n)$. As before, (1) highlight the non-recursive parts and estimate their runtime, then (2) highlight the recursive parts and estimate their runtime using $T(n)$, then (3) finally evaluate $T(n)$ using the Master Theorem.

```
Function F(array A[1...n])
    if n = 1
     then
        return A[1]
    end
    perform $O(n)$ steps to compute arrays B, C and D, of size $\lfloor n/3 \rfloor$ each
    x = F(B)
    y = F(C)
    z = F(D)
    return x+y+z
```

```
Function G(array A[1...n])
    if n = 1
     then
        return A[1]
    end
    perform $O(n)$ steps to compute arrays B, C and D, of size $\lfloor n/3 \rfloor$ each
    x = G(B)
    y = G(C)
    return x-y
```

```
Function H(array A[1...n])
    if n = 1
     then
        return A[1]
    end
    perform $O(n)$ steps to compute arrays B, C, D, E, of size $\lfloor n/3 \rfloor$ each
    x = H(B)
    y = H(C)
    z = H(D)
    w = H(E)
    return x+y+z+w
```

```
Function I(array A[1...n])
    if n = 1
     then
        return A[1]
    end
    perform $O(n^2)$ steps to compute arrays B, C, D, E, of size $\lfloor n/2 \rfloor$ each
    x = I(B)
    y = I(C)
    z = I(D)
    w = I(E)
    return x+y+z+w
```