**CSCI\_1471 Worksheet                                      Student Name: Geneiva Ocampo**

**Worksheet is due on Blackboard 11:55pm, Thursday, 25 Apr 24.**

**Recursion**

**What does Recursion Require?**

1. Termination Condition
2. Call the Clone (same function)
3. Move towards Termination Condition

Label the Recursion Requirements.                                                               Show the Activation Stack for n = 4.

    int fact(int n)

    {        int result;

             if(n==1)

               return 1;

            result = fact(n-1) \* n;

           return result;

       }//end method

boolean isPal(String s)                           **Show the Activation Stack for “ZEROREZ”**

 {

     if(s.length() == 0 || s.length() == 1)

         // if length =0 OR 1 then it is

         return true;

     if(s.charAt(0) == s.charAt(s.length()-1))

         // check first and last char of String:

         // if same then do same for substring

         // for substring and carry on this

         // remove first and last char.

         return isPal(s.substring(1, s.length()-1));

     // if its not the case than string is not.

     return false;

 }

**Big O**

  -  Big O is an algebra that permits us to express how the amount of work done in a problem relates to the amount of data.

  -  Basic Elements: Big O is the worst case performance

            - O(1)

                        - O(logN)

                        - O(N)

                        - O(N^2) - Bubble Sort

                        - O(2^N) – Fibonacci

-  Big O is determined by the worst-case performance of an operation.

- There are simplification rules with Big O.

* no constant multipliers
* keep only term of fastest growth rate.

- Multiple operations are often required to solve problems (Complex algorithms).

* if sequential,
* if nested, multiply
* Examples: O(NlogN) ; O(N+M); O(N\*M)

**Question:   An average performance that would claim (N+1)/2 performance what is the BigO?**

**Answer**:  We will reject the “1” as being N to a lower power and “2” as a constant multiplier.  This leaves O(N).

**Question:  If the three operations were combined, O(logN) + O(N) \* O(logN) +1, the overall algorithm cost would be:**

1. O(N)
2. O(logN)
3. O(logN) + O(N)
4. O(NlogN)

            What is the dominate term of each function?

* *3n2+4n-3*

* *7n3+ 4n2-7n + 5*

* *5nlog(n) + 2n – 4*

**Sorting Algorithms**

%   - algorithms only - no code

%   - performance comparison – sort demo

%   - uses for each algorithm

http://www.sorting-algorithms.com/random-initial-order

**Sorting Algorithms**

|  |  |  |  |
| --- | --- | --- | --- |
| Type Sort | Big "O" | Form | Use |
| Insertion | N^2 | Single Loop | Used when adding to sorted data |
| Bubble | N^2 | Nested For Loops | Used for sorting short vectors, in place |
| Merge | NlogN | Recursive  Recursive  Merge | Used for merging sorted data |
| Quick | NlogN | Partition  Recursive  Recursive | Used for random data, not good for sorted. |

**% Insertion Sort (Efficient for growing with time)**

    [55    96    97    16    98    96    49]

**%   Bubble Sort  (Good for short vectors in place)**

      [81    15    43    92    80    96    6]

**%   Merge Sort  (good for sorted data)**

    [ 4    85    94    68    76    75    40    66    18]

**%   Quick** **Sort  (not good if data is already sorted)**

    [47    44    39    77    80    19    49    45    65]

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
| Main() | isPal(“ZEROREZ”) |

Method Boolean inPal(String s)

Terminating Conditions 🡪if(s.lenght =0||s.lenght()==1)true, if(s.lenght =0||s.lenght()-11)true