Name - Prabhas Kumra

Assignment #2

CS 302 – 1004

1. Formal definitions:
    a. Big Oh – Notation for upper bound (worst case) to its growth rate of $f(n)$

        $T(N) = O\big(f(N)\big)$, if there are positive constant C and no Such that

        $T(N) \leq C$.

    b. Big Ω - Notation for lower bound (best case) to its growth rate of $f(n)$

        $T(N) = \Omega\big(g(N)\big)$, if there are positive constants C and no such that

        $T(N) \leq C.g(N), when\ N \geq n$

    c. Big Θ - Notation for, when the upper bound and lower bound are the same within a constant factor.

        $T(N) = \Theta\big(n(N)\big)$, if and only if $T(N) = \mathrm{O}\big(h(N)\big)$

        when $T(N) = \Omega\big(h(N)\big)$
2. Shor informal description:
    a. Big Oh – Notation for upper bound (worst case) to its growth rate of $f(n)$
    b. Big Ω - Notation for lower bound (best case) to its growth rate of $f(n)$
    c. Big Θ - Notation for, when the upper bound and lower bound are the same within a constant factor.
3. Growth rate order: $\frac{2}{n}, \log n, \sqrt{n}, n \log n, 2n, 2^{\frac{n}{2}}, n^2 \log n, 2^n, n^3, 4n^2, 730$
4.
    a. Big O analysis for binary search for finding an element in sorted array is $O(\log n)$
    b. Because in Binary search there are $O(\log n)$ comparisons, where *n* is the number of elements in the array. For sequential search Big-Oh is $O(n)$, where *n* is the number of elements.
5. Big-Oh for inserting a new element in an unsorted linked list is $O(1)$
6. Big-Oh for inserting a new element in sorted linked list is $O(1)$
7. Big-Oh of an algorithm to find if a number is prime is $O(N)$

8. –
    a. $O(n^2)$
    b. $O(n^2)$
9. –
    a. $O(n \log n)$
    b. $O(n)$
10. –
    a. $O(n^2)$
    b. $O(n \log n)$
    c. $O(n)$
    d. $O(n^2)$
    e. $O(n)$
    f. $O(n^3)$
    g. $O(n^4)$
    h. $O(n^3)$
    i. $O(n^2)$
11. –
    a. Algorithm 1 - $O(n)$
    b. Algorithm 2 - $O(n)$
12. –
    a. Algorithm 1 - $O(n)$
    b. Algorithm 2 - $O(2^n)$
13. –
    a. Algorithm 1 - $O(2^n)$
    b. Algorithm 2 - $O(n)$
14. –
    a. The space/time tradeoff principle says that one can often achieve a reduction in time if one is willing to sacrifice space or vice versa.
    b. Example of space/time tradeoff is a lookup table.

15. –

a. Big-Oh for recursive algorithm in assignment #1 was, $O(2^n)$
b. Big-Oh for dynamic algorithm in assignment #1 was, $O(n^2)$