

CS-285
Lab 2
Algorithm Analysis

Analyze the worst-case time complexity of the following recursive algorithms in “Big-Oh” notation using recurrence relations:

// 1) Given a linked list of n nodes, the following prints out the list

```
void print(Node* ptr)
{
    if (ptr != NULL)
    {
        cout << ptr->data;
        print(ptr->next);
    }
}
```

// 2) Analyze the following function:

```
int recursive(unsigned int n)
{
    if (n <= 1)
        return n;
    else
        return (recursive(n-1) + recursive(n-1));
}
```

3) Order the following time complexity functions in best to worst order:

$$n, \sqrt{n}, \log n, \log \log n, \log^2 n, \frac{n}{\log n}, n \log n, \left(\frac{1}{2}\right)^n, \left(\frac{3}{2}\right)^n, 2^{10}$$