COP 3502 Study Group Sheet: Algorithm Analysis and Linked Lists

Directions: These are questions from an old exam. Based on the quiz 2 results, not only will it be good for many to practice the current course material, but it would be very useful to go back and get extra practice with recursion and linked list. These questions provide practice on both of these topics together.

1) An $O(n^2)$ algorithm takes 2 ms. to run on an input size of n = 50. On another input size, the algorithm took 72 ms. to run, what was that input size?

2) Write a function that takes in a pointer to the front of a linked list and an integer threshold and returns the number of values in the linked list greater than threshold.

```
struct node {
    int data;
    struct node* next;
};

int countBigger(struct node* front, int threshold) {









}
```

3) What is the run-time (Big-Oh) in terms of n of the following code segments? (Assume all variables have been previously declared as integers.)

a)
```
int i, j, sum = 0;
for (i=0; i<n; i++) {
  for (j=0; j<i; j++) {
    if (j > i)
      sum++;
  }
}
```

_____

b)
```
int i, sum = 0, saven = n;
while (n > 0) {
  for (i=0; i<saven; i++)
    sum++;
  n = n/2;
}
```

_____

c)
```
int sum = 0;
while (n > 0) {
  sum++;
  n = n/3;
}
```

_____

4) Determine an expression in terms of n for the two following sums:

a) $\sum_{i=n+1}^{2n}(3i-1)$

b) $\sum_{i=7}^{n}8$

5) Use the iteration technique to solve the following recurrence:

$T(n) = T(n/2) + n, T(1) = 1$

(Hint: In solving this recurrence utilize the fact that $\sum_{i=0}^{k} \frac{1}{2^i} \approx 2$ for large values of k.)

6) Write a recursive function that returns the largest value in a linked list of positive integer values. If the list passed to the function is null, return 0.

```
struct node {
    int data;
    struct node* next;
};

int maxVal(struct node* front) {









}
```

7) Write a function that returns 1 if a linked list is in ascending sorted order with no duplicates and 0 otherwise. If the list is empty, your function should return 1.

```
struct node {
    int data;
    struct node* next;
};

int inOrder(struct node* front) {









}
```