CMPSC 463: Problem Set #4

For each of the following recurrence relations, solve the relation using the Master Theorem, or write does not apply if the Master Theorem does not apply. Show your work. (5 points each)

1) $T(n) = T\left(\frac{n}{2}\right) + n$
2) $T(n) = 2nT\left(\frac{n}{2}\right) + n$
3) $T(n) = 5T\left(\frac{n}{2}\right) + n$
4) $T(n) = T(n) + c$ (where c is a constant)
5) $T(n) = 4T\left(\frac{n}{4}\right) + O(1)$
6) $T(n) = 3T\left(\frac{n}{4}\right) + 1$
7) $T(n) = 4T\left(\frac{n}{2}\right) + n\log n$
8) $T(n) = 4T\left(\frac{n}{2}\right) + n^2$
9) $T(n) = 2T\left(\frac{n}{2}\right) + 2^n$

For each of the following recurrence relations, solve the relation. Show your work. (5 points each)

10) $T(n) = T\left(\frac{n}{3}\right) + T\left(\frac{n}{4}\right) + n$
11) $T(n) = T\left(\frac{n}{2}\right) + 2^n$
12) $T(n) = T(n-4) + O(1)$

Write recurrence relation for the worst-case time complexity for the next two algorithms provided below. If possible, solve the relation using the Master Theorem. Show your work. (20 points each)

13) (Note: in top level call s = 1 and t = n)

```
searchForValue(A[1..n], s, t, k)
    if (s > t) return -1

    m = floor(( 3 * s + t ) / 4)// Divide array into two chunks, one
                                // with 1/4 of the elements and the
                                // other 3/4 of the elements
    if (A[m] = k)
        return m
    else if (A[m] > k)
        return searchForValue(A,m+1,t,k)
    else
        return searchForValue(A,s,m-1,k)
    end if
end searchForValue
```

14) The following algorithm takes as input a heap A (arranged as an array), and an index of a root, and returns the sum of the elements in the heap.

(Note: in top level call r = 1)

```
sum(A[1..n], r)
    if r > n then return 0

    return A[r] + sum(A, 2*r) + sum(A, 2*r + 1)
end sum
```