(a) Pseudocode for MOM7QS

```
// sorts A where A is a global array
QuickSort (A) :
    n = |A|
    MOM7QS (1,n)

// MOM7QS (low, high) : sorts A[low... high] using modified quicksort

MOM7QS (low, high) :
    if (low < high) :
        mom = computeMom (low, high)
        mid = partition ( low , high)
        MOM7QS (low, mid-1)
        MOM7QS (mid+1 , high)

// computeMom (l,h) : selects the pivot (for partitioning A) by
                      computing median of 7 medians of A[l...h]

// assumes mom is a global variable

computeMom (l,h) :
    if n <= 49 :
        return median (l, h)
    m = ⌈n/7⌉
    for i ← 1 to m
        M[i] = median (7i-6, 7i)
    return computeMom (1,m)

// median (l,h) : computes the median of unsorted A[l..h]

median (l,h) :
    sorts A in the range l to h
    i = h-l+1
    if (i % 2 != 0):
        return A[l + (i/2)]
    else return A[l + ((i/2)-1)]
```

(b) Recursive formula to compute T(n) - worst case complexity of MOM7QS

Let $Q(n)$ denote the time complexity of QuickSort(A)

Let $T(n)$ denote the time complexity of MOM7QS()

Let $C(n)$ denote the time complexity of computeMom()

Let $P(n)$ denote the time complexity of partition()

Let $M(n)$ denote the time complexity of median()

$$Q(n) = T(n)$$

$$T(n) = \underbrace{C(n)}_{\text{from line 3}} + \underbrace{P(n)}_{\text{from line 4}} + \underbrace{\max_{r=1\ldots n} \{ \cancel{T(r)}\; T(r-1) + T(n-r) \}}_{\text{from line 5, 6}}$$

$$C(n) = \underbrace{O(1)}_{\text{from line 2}} + \underbrace{O(1)}_{\text{from line 3}} + \underbrace{\left\lceil \tfrac{n}{7} \right\rceil * O(1)}_{\text{from line 4,5}} + \underbrace{C\left(\left\lceil \tfrac{n}{7} \right\rceil\right)}_{\text{from line 6}}$$

$$C(n) \cong C\left(\left\lceil \tfrac{n}{7} \right\rceil\right) + O(n)$$

$P(n) = O(n)$ (We know that time complexity of partition() of standard quicksort algorithm is $O(n)$)

$M(n) = O(c)$ (where c is a const.)

time complexity of line 3,4,5 in median() is $O(1)$ clearly.

Now, in line 2, best case time complexity will be $M(n) = 7\log 7$ and worst case time complexity will be $M(n) = 49 \log 49$ both of these are constant values.

Therefore $M(n) = O(c)$

(c) Solveing $T(n)$

$C(n) = C\left(\frac{n}{7}\right) + O(n)$ {ignoring the ceil}

here using master's theorem we get $C(n) = \Theta(n)$

Now replacing values of $C(n)$ and $P(n)$ in the reccurence of $T(n)$ we get

$$T(n) = \max_{r=1\ldots n} \left\{ \cancel{T(r) + T(n-r)} \atop \{T(r-1) + T(n-r)\} \right\} + O(n)$$

in case of this algorithm $\frac{2n}{7} \le r \le \frac{5n}{7}$

$$T(n) \le T\left(\frac{2n}{7}\right) + T\left(\frac{5n}{7}\right) + O(n)$$

Guessing that $T(n) \le cn\log n$

To show: $cn\log n \le \frac{2cn}{7}\log\left(\frac{2n}{7}\right) + \frac{5cn}{7}\log\left(\frac{5n}{7}\right) + dn$

let $d = c$

$$cn\log n \le \frac{2cn}{7}\log\left(\frac{2n}{7}\right) + \frac{5cn}{7}\log\left(\frac{5n}{7}\right) + cn$$

$\log n \le \frac{2}{7}\log\left(\frac{2n}{7}\right) + \frac{5}{7}\log\left(\frac{5n}{7}\right) + 1$ (cancelling cn from both sides)

$$\log n \le \frac{2}{7}\log\left(\frac{n}{7/2}\right) + \frac{5}{7}\log\left(\frac{n}{7/5}\right) + 1$$

$$\log n \le (\log n) * \left(\frac{2}{7} + \frac{5}{7}\right) + \left(1 - \frac{2}{7}\log 7/2 - \frac{5}{7}\log 7/5\right)$$

$$\log n \le \log n + (1 - 0.52 - 0.35)$$

$0 \le 0.13$ (which is trivial, hence our goal is proved)

Therefore $T(n) = O(n\log n)$