Practice Assignment 4

Andrew Johnson

1) a) $T(n) = 2T(n/3) + 1 = \sum^{\log_3 n} c\left(\frac{n}{3^i}\right)^0 \cdot 2^i = c\Theta(2^{\log_3 n}) = \Theta(n)$

b) $T(n) = 5T(n/4) + n = \sum^{\log_4 n} c\left(\frac{n}{4^i}\right)5^i = cn\sum^{\log_4 n}\left(\frac{5}{4}\right)^i = \Theta\left(cn\left(\frac{5^{\log_4 n}}{n^2}\right)\right)$

$= \Theta\left(\frac{n^{\log_4 5}}{n}\right) = \Theta\left(n^{(\log_4 5 - 1)}\right)$

c) $T(n) = 7T(n/7) + n = \sum^{\log_7 n} c\left(\cancel{n/7}\right)\cancel{7^i} = \Theta(n\log n)$

d) $T(n) = 9T(n/3) + n^2 = \sum^{\log_3 n} c\left(\frac{n}{3^i}\right)^2 9^i = \sum^{\log_3 n} cn^2 = \Theta(n^2\log n)$

e) $T(n) = 8T(n/2) + n^3 = \sum^{\log_2 n} c\left(\frac{n}{2^i}\right)^3 8^i = \sum^{\log_2 n} cn^3 = \Theta(n^3\log n)$

f) $T(n) = T(n-1) + 2 = T(n-k) + 2\cdot k \;\; \forall k \Rightarrow T(n-n) + 2n \qquad T(0) = \Theta(1)$

$= T(0) + 2n = \Theta(n)$

g) $T(n) = T(n-1) + C^n$, for $c > 1 \Rightarrow T(n-n) + nC^n = \Theta(nc^n) \quad c > 1$

h) $T(n) = 2T(n-1) + 1 = 2(2^{n-1} - 1) + 1 = 2^n - 1 = \Theta(2^n)$

i) $T(n) = 49T(n/25) + n^{3/2}\log n = \sum^{\log_{25} n} c\left(\frac{n}{25^i}\right)^{3/2}\log(n/25)\,49^i$

$= cn^{3/2}\left[\log(n) - \log 25\right]\sum^{\log_{25} n}\left(\frac{49}{125}\right)^i$

$= n^{3/2}\log(n)\left(\frac{49}{125}\right)^{\log_{25} n}$

j) $T(n) = T(n-1) + n^c$, for $c \geq 1 \Rightarrow T(n-n) + n\cdot n^c \Rightarrow \Theta(n^{c+1}) \quad c \geq 1$

K) $T(n) = T(\sqrt{n}) + 1$, Let $n = 2^{2^m}$ so $\sqrt{n} = 2^{2^{m-1}}$, looking at n going from $2^{2^0} \to 2^{2^m}$, we see it's incremented by m eachtime, thus $T(n) = m$ and $m = \log_2(\log_2(n))$ since $n = 2^{2^m}$,

so we get $\Theta(\log(\log(n)))$

2) Lets take $n=1$ so our first set is $\{n\}$ and second set is $\{n+1\}$, which is $\{1\}$ and $\{2\}$ respectively. We can notice these sets don't have any matching factors, thus the induction can't prove that these two sets have the same colored horses. ■

3)
```
GetMedian (a[1,...,n], b[1,...,n])
    if (n ≤ 0)
        return -1
    if (n = 1)
        return (a[1] + b[1]) / 2
    if (n = 2)
        return (max(a[1], b[1]) + min(a[2], b[2])) / 2

    m_a = median(a, n)
    m_b = median(b, n)

    if (m_a = m_b)
        return m_a
    if (m_a < m_b)
        if (n % 2 = 0)
            return GetMedian(a[n/2-1, n], b[1, n/2+1]
        else
            return GetMedian(a[n/2, n], b[1, n/2])
    else
        if (n % 2 = 0)
            return GetMedian(a[1, n/2+1], b[n/2-1, n])
        else
            return GetMedian(a[1, n/2], b[n/2, n])
```

```
median(a[1,..,n])
  if (n%2 = 0)
     return (a[n/2]-a[n/2-1])/2
  else
     return a[n/2]
```

$T(n) = T(\frac{n}{2}) + 1 = \sum^{\log_2 n} c(\frac{n}{2^i})^0 = c\log_2 n = \Theta(\log n)$

4) a) Take an array
Base Case (if size is 2)
swap if needed
if size > 2
get 2/3 of the size of the array
recursively call first 2/3 of array
recursively call last 2/3 of array
recursively call first 2/3 of array

It takes the large numbers from the first 2/3's and sets them in the middle, then it takes the last two thirds and sets the larger numbers in the last third and smaller in the middle third. Lastly, it'll take any small numbers that were moved to the middle third and place them in the first third if needed.

b) Using the floor with an array of size 4, when spliting the array it'll look at a[1,2] and a[3,4] thus the arrays never overlap and can't switch their numbers with eachother.

c) $T(n) = 3T(\frac{2n}{3}) + \Theta(1)$

d) $\sum_{i=1}^{\log_2 n} c(\frac{2n}{3})^0 3^i \Rightarrow \Theta(3^{\log_2 n}) \Rightarrow \Theta(n^{\log_2 3})$