Handwriting Assignment #1 Solution : Algorithm

1. Prove by induction that $2^n < n!$ for every integer $n \geq 4$.

Solution) Base case: $2^4 = 16$, and $4! = 24$, so $2^4 < 4!$.

Induction step: Let integer $k \geq 4$, and assume that $2^k < k!$.
We wish to show that $2^{k+1} < (k + 1)!$.
We have $2^{k+1} = 2 \cdot 2^k < 2 \cdot k!$ by the induction hypothesis.
But then we also have $2 < (k + 1)$ since $k \geq 4$.
Therefore $2^{k+1} < 2 \cdot k! < (k+1) \cdot k! = (k + 1)!$, as desired.

2. Decide whether each of the following statements is true or false, and prove that your conclusion is correct.

1) $n^2 = O(2^n)$

True, since $n^2 \leq c\,(2^n)$ for all $n_0 \geq 2$ and $c = 1$, for example.

2) $2^{n+1} = O(2^n)$

True, since $2^{n+1} = 2 \cdot (2n)$, so we can pick for example $n0 = 1$ and $c = 2$

3) $2^{2n} = O(2^n)$

False, since $2^{2n} = (2^n)^2$ and this cannot be bound from above by a constant.

3. Solve the following recurrence using the master method. Show your work.

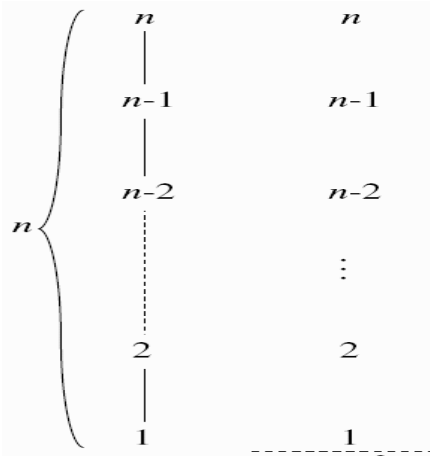
$$T(n) = \begin{cases} \Theta(1) & n = 1 \\ T(n/2) + \Theta(1)) & n > 1 \end{cases}$$

Master method: $a = 1$, $b = 2$,
compare $f(n)$ $n^{\log_b a}$
$n^{\log_b a} = n^0$
Therefore, we are in case 2 of the master theorem, and $T(n) = \Theta(\log n)$

4. Solve the recurrence by recursion tree

$$T(n) = \begin{cases} 1 & n = 1 \\ T(n-1) + n & n > 1 \end{cases}$$

Solution) Levels Cost



$n(n+1)/2$

$n + n-1 + n-2 + \ldots + 2 + 1 =$

$$\sum_{h=1}^{n} h = \frac{n(n+1)}{2} = \Theta(n^2)$$

5.Solve the following recurrence by the master method.

$$T(n) = T(\frac{9n}{10}) + n.$$

$$T(n) = T(\frac{9n}{10}) + n.$$

This is a divide-and-conquer recurrence with $a = 1$, $b = 10/9$, $f(n) = n$, and $n^{\log_b a} = n^{\log_{10/9} 1} = n^0 = 1$. This is case 3 of the master theorem, thus $T(n) \in \Theta(n)$.

6. Trace the insertion sort as it sorts the following array into ascending order:
25 30 20 80 40 60

Solution:

```
25 30 20 80 40 60
25 30 20 80 40 60
20 25 30 80 40 60
20 25 30 40 80 60
20 25 30 40 60 80
```

7. Trace the *mergesort* algorithm as it sorts the following array into ascending order. List the calls to *mergesort* and *merge* in the order in which they occur.
20 80 40 25 60 30

```
(1) mergesort(0,5)
(2)   mergesort(0,2)
(3)     mergesort(0,1)
(4)       mergesort(0,0)
(5)       mergesort(1,1)
(6)       merge(0,0,1)
(7)     mergesort(2,2)
(8)     merge(0,1,2)
(9)   mergesort(3,5)
(10)    mergesort(3,4)
(11)      mergesort(3,3)
(12)      mergesort(4,4)
```