HW #1

1. Order the following functions according to their order of growth (from the lowest to the highest). If any two or more are of same order, indicate which.

$f_1(n) = n^2 + \log n$ $\quad$ $f_8(n) = n^{12} + n^{10}$

$f_2(n) = \sqrt{n}$ $\quad$ $f_9(n) = n^{12} \log n$

$f_3(n) = n - 1000$ $\quad$ $f_{10}(n) = n^{\frac{1}{3}} + \log n$

$f_4(n) = n \log n$ $\quad$ $f_{11}(n) = (\log n)^2$

$f_5(n) = 2^n + n^{10}$ $\quad$ $f_{12}(n) = 10^{15}$

$f_6(n) = n^5 + 3^n$ $\quad$ $f_{13}(n) = \dfrac{n}{\log n}$

$f_7(n) = n^{11}.2^{2\log n}$ $\quad$ $f_{14}(n) = \log\log n$

2. What value is returned by the following algorithm? What is its basic operation? How many times is the basic operation executed? Give the worst-case running time of the algorithm using Big Oh notation.

```
MICHIGAN(n)
input : an integer n
r ← 0
for i = 1 to n
    for j = i + 1 to n
        for k = i + j − 1 to n
            r ← r + 1
return r
```

3. Solve the following recurrence relation using recursion tree method.

$$T(n) = \begin{cases} 1 & , \text{ if } n \le 2 \\ 4T\left(\frac{n}{2}\right) + n^2, & \text{if } n > 2 \end{cases}$$

4. Solve the following recurrence relation using recursion tree method.

$$T(n) = \begin{cases} 1 & , \quad \text{if } n \le 2 \\ T\left(\frac{n}{3}\right) + T\left(\frac{2n}{3}\right) + n, & \text{if } n > 2 \end{cases}$$

5. What does the following algorithm compute? What is its basic operation? How many

times is the basic operation executed? Give the worst-case running time of the algorithm using Big Oh notation.

ALASKA(A = $(a_{ij})_{nxn}$)
input : an nxn matrix of real numbers
r ← 0
for i = 1 to n-2
    for j = i + 1 to n
        if $a_{ij} \neq a_{ji}$
            return false
return true

6. Solve the following recurrence relation using Master Theorem.

$$T(n) = \begin{cases} 1 & , \quad \text{if } n \leq 2 \\ 2T\left(\frac{n}{2}\right) + n\log n, & \text{if } n > 2 \end{cases}$$

7. Solve the following recurrence relation using Master Theorem.

$$T(n) = \begin{cases} 1 & , \quad \text{if } n \leq 2 \\ 3T\left(\frac{n}{3}\right) + \sqrt{n}, & \text{if } n > 2 \end{cases}$$

8. What does the following recursive algorithm compute? Set up a recurrence relation for the running time of the algorithm and solve it using backward substitution.

SAMSUN(a_i, a_{i+1}, …, a_j)
input : a sequence of integers
if $i = j$
  return a_i
else
  mid ← $\lfloor (i + j)/2 \rfloor$
  temp1 ← **SAMSUN(a_i, …, a_{mid})**
  temp2 ← **SAMSUN(a_{mid}, …, a_j)**
  if temp1 ≤ temp2
    return temp1
  else
    return temp2