HW#3 Solution, CSCI 323, Spring 2003
Department of Computer Science
Queens College
Dr. Song

Problem 1 Solution. G&T R-2.14

- Pre-order: Yes.For example, a pre-order traversal of the following 7 node heap will result a visit of the nodes in an ascending order: 1, 2, 3, 4, 5, 6, 7.

```
      1
    /   \
  2       5
 / \     / \
3   4   6   7
```

- In-order: No. In a seven node heap shown below, an in-order traversal will result in a visit of the nodes in the order: d, b, e, a, f, c, g. By heap order property we know that $d > b < e$, so the list is not sorted.

```
      a
    /   \
  b       c
 / \     / \
d   e   f   g
```

- Post-order: Yes.For example, a pre-order traversal of the following 7 node heap will result a visit of the nodes in a descending order: 7,6,5,4,3,2,1.

```
      1
    /   \
  5       2
 / \     / \
7   6   4   3
```

Problem 2 Solution. G&T R-2.18

The original heap can be:

```
                               1
                /                             \
               3                              33
        /             \                /              \
       5               7              35              37
    /     \         /     \         /     \         /     \
   9      11      13      15      39      41      43      45
 /  \    /  \    /  \    /  \    /  \    /  \    /  \    /
17  19  21  23  25  27  29  31  47  49  51  53  55  57  59
```

After insertion of 32, the heap becomes:

```
                               1
                /                             \
               3                              32
        /             \                /              \
       5               7              35              33
    /     \         /     \         /     \         /     \
   9      11      13      15      39      41      43      37
 /  \    /  \    /  \    /  \    /  \    /  \    /  \    /  \
17  19  21  23  25  27  29  31  47  49  51  53  55  57  59  45
```

Problem 3 Solution. G&T R-4.3

Algorithm 1 MergeSort(A)

INPUT: an array A of size n.
OUTPUT: a sorted array D of A in ascending order
Copy the first $L_1 = \lceil n/2 \rceil$ elements in A to A_1
Copy the remaining $L_2 = \lfloor n/2 \rfloor$ elements in A to A_2
$D_1 \leftarrow$ MergeSort(A_1)
$D_2 \leftarrow$ MergeSort(A_2)
Create an array D of size n
$i_1 \leftarrow 0$
$i_2 \leftarrow 0$
$i \leftarrow 0$
while $i_1 < L_1$ and $i_2 < L_2$ **do**
  if $D_1[i_1] < D_2[i_2]$ **then**
    $D[i] \leftarrow D_1[i_1]$
    $i_1 \leftarrow i_1 + 1$
  else
    $D[i] \leftarrow D_2[i_2]$
    $i_2 \leftarrow i_2 + 1$
  end if
  $i \leftarrow i + 1$
end while
while $i_1 < L_1$ **do**
  $D[i] \leftarrow D_1[i_1]$
  $i_1 \leftarrow i_1 + 1$
  $i \leftarrow i + 1$
end while
while $i_2 < L_2$ **do**
  $D[i] \leftarrow D_2[i_2]$
  $i_2 \leftarrow i_2 + 1$
  $i \leftarrow i + 1$
end while
return D

Problem 4 Solution. G&T R-5.4

1. $T(n) = 2T(n/2) + \log n$
   Case 1. $\Theta(n)$.
2. $T(n) = 8T(n/2) + n^2$.
   Case 1. $\Theta(n^3)$.

3. $T(n) = 16T(n/2) + (n \log n)^4$
   Case 2. $\Theta(n^4 \log^5 n)$.
4. $T(n) = 7T(n/3) + n$
   Case 1. $\Theta(n^{\log_3 7})$
5. $T(n) = 9T(n/3) + n^3 \log n$
   Case 3. $\Theta(n^3 \log n)$.

Problem 5 Solution. G&T R-5.6

$$
\begin{aligned}
S_1 &= A(F-H) = 3(5-6) = -3 \\
S_2 &= (A+B)H = (3+2)6 = 30 \\
S_3 &= (C+D)E = (4+8)1 = 12 \\
S_4 &= D(G-E) = 8(9-1) = 64 \\
S_5 &= (A+D)(E+H) = (3+8)(1+6) = 77 \\
S_6 &= (B-D)(G+H) = (2-8)(9+6) = -90 \\
S_7 &= (A-C)(E+F) = (3-4)(1+5) = -6 \\
I = S_5 + S_6 + S_4 - S_2 &= 77 + -90 + 64 - 30 = 21 \\
J = S_1 + S_2 &= -3 + 30 = 27 \\
K = S_3 + S_4 &= 12 + 64 = 76 \\
L = S_1 - S_7 - S_3 + S_5 &= -3 - (-6) - 12 + 77 = 68
\end{aligned}
$$

Problem 6 Solution.

1. (a) $T(n) = 2T(n/2) + \log n$

$$
\begin{array}{cc}
k=0, & T(n) = 2T(n/2) + \log n \\
k=1, & 2^1 T(n/2^1) = 2^2 T(n/2^2) + 2^1 \log(n/2^1) \\
\cdots & \\
k, & 2^k T(n/2^k) = 2^{k+1} T(n/2^{k+1}) + 2^k \log(n/2^k) \\
\cdots & \\
k = \log_2(n) - 1, & \ldots \\
k = \log_2(n), & 2^{\log_2(n)} T(1) = 2^{\log_2(n)} \\
\hline
\end{array}
$$

$$T(n) = 2^{\log_2(n)} + \sum_{k=0}^{\log_2(n)-1} 2^k \log(n/2^k)$$

By plugging in the geometric sequence summation formula and the following formula

$$\sum_{k=1}^{n} k a^k = \frac{a(1-a^n)}{(1-a)^2} - \frac{n a^{n+1}}{1-a}, \qquad (a \neq 1)$$

we get

$$T(n) = 3n - \log n - 2$$

(b) $T(n) = 8T(n/2) + n^2$.

$$\begin{array}{cc}
k=0, & T(n) = 8T(n/2) + n^2 \\
k=1, & 8^1 T(n/2^1) = 8^2 T(n/2^2) + 8^1 (n/2^1)^2 \\
\cdots & \\
k, & 8^k T(n/2^k) = 8^{k+1} T(n/2^{k+1}) + 8^k (n/2^k)^2 \\
\cdots & \\
k = \log_2(n) - 1, & \ldots \\
k = \log_2(n), & 8^{\log_2(n)} T(1) = 8^{\log_2(n)} \\
\hline
\end{array}$$

$$T(n) = 8^{\log_2(n)} + \sum_{k=0}^{\log_2(n)-1} 8^k (n/2^k)^2$$

By plugging in the geometric sequence summation formula, we get

$$T(n) = 2n^3 - n^2$$

(c) $T(n) = 7T(n/3) + n$.

$$\begin{array}{cc}
k=0, & T(n) = 7T(n/3) + n \\
k=1, & 7^1 T(n/3^1) = 7^2 T(n/3^2) + 7^1 n/3^1 \\
\cdots & \\
k, & 7^k T(n/3^k) = 7^{k+1} T(n/3^{k+1}) + 7^k n/3^k \\
\cdots & \\
k = \log_3(n) - 1, & \ldots \\
k = \log_3(n), & 7^{\log_3(n)} T(1) = 7^{\log_3(n)} \\
\hline
\end{array}$$

$$T(n) = 7^{\log_3(n)} + \sum_{k=0}^{\log_3(n)-1} 7^k n/3^k$$

By plugging in the geometric sequence summation formula, we get

$$T(n) = \frac{7}{4} n^{\log_3 7} - \frac{3}{4} n$$

.

2. Proof (by induction)

Need to show:

- $T(n)$ is $O(n^3 \log n)$.
  Proof.
  Since we do not know what $c > 0$ to take, we start from the induction step.
  Assume that $T(n') \leq c n'^3 \log n'$ for $n' < n$.

$$\begin{aligned}
T(n) &= 9T(n/3) + n^3 \log n \\
&\leq 9c[(n/3)^3 \log(n/3)] + n^3 \log n \\
&= ((c/3) + 1) n^3 \log n - (c/3) n^3 \log 3 \\
&\leq ((c/3) + 1) n^3 \log n
\end{aligned}$$

To satisfy $T(n) \leq cn^3 \log n$, it is sufficient that

$$((c/3)+1)n^3 \log n \leq cn^3 \log n$$

which leads to $c \geq 3/2$.

Base case. Since $T(1) = 1 \leq c \cdot 1$ for any $c \geq 3/2$, we have no problem to use the c that is sufficient for the induction step.

Therefore, $T(n)$ is $O(n^3 \log n)$.

- $T(n)$ is $\Omega(n^3 \log n)$.

  Proof.

  We still start from the induction step. Assume that $T(n') \geq cn'^3 \log n'$ for $n' < n$.

$$\begin{aligned} T(n) &= 9T(n/3) + n^3 \log n \\ &\geq 9c[(n/3)^3 \log(n/3)] + n^3 \log n \\ &\geq n^3 \log n \qquad \text{for } n \geq 3\text{, since the first term in the above line is non-negative when } n \geq 3 \end{aligned}$$

  To satisfy $T(n) \geq cn^3 \log n$, it is sufficient that

$$n^3 \log n \geq cn^3 \log n$$

  which leads to $c \leq 1$.

  Base case. Induction step requires $n \geq 3$. We have

$$T(1) = 1, T(3) = 9T(3/3) + 3^3 \log 3 = 9 + 3^3 \log 3$$

  Since $T(3) \geq c \cdot 3^3 \log 3$ for any $c \leq 1$, we have no problem to use the c that is sufficient for the induction step.

  Therefore, $T(n)$ is $\Omega(n^3 \log n)$.