Homework 5 Solutions

Problem 1:

(1) The critical value is $c = \log_3(4) \approx 1.262$. We need to compare n^c with $\log(n)n$. Set $d = c - 1 - \epsilon$ with $\epsilon < 0.2$ so that $0 < d < 1$. $\lim_{n\to\infty} \frac{n^{c-\epsilon}}{n\log(n)} = \lim_{n\to\infty} \frac{n^{1+d}}{n\log(n)}$ $= \lim_{n\to\infty} \frac{n^d}{\log(n)} =^{LH} \lim_{n\to\infty} \frac{dn^{d-1}}{1/n} = \lim_{n\to\infty} dn^d = \infty$. Thus, $f(n) = O(n^{c-\epsilon})$ and therefore $T(n) = \Theta(n^c)$.

(2) The critical value is $c = \log_3(3) = 1$. $f(n) = \sqrt{n} = O(n^{1-\epsilon})$ for any $\epsilon < \frac{1}{2}$, so $T(n) = \Theta(n)$.

(3) The critical value is $c = \log_2(2) = 1$. Since $\lim_{n\to\infty} \frac{n\log(n)}{n} = \lim_{n\to\infty} \log(n) = \infty$, $f(n) = n\log(n) \notin O(n)$. However, if $\epsilon > 0$, then

$$\lim_{n\to\infty} \frac{n^{1+\epsilon}}{n\log(n)} = \lim_{n\to\infty} \frac{n^\epsilon}{\log(n)} =^{LH} \lim_{n\to\infty} \frac{\epsilon n^{\epsilon-1}}{\frac{1}{n}} = \lim_{n\to\infty} \epsilon n^\epsilon = \infty$$

so that $f(n) \notin \Omega n^{1+\epsilon}$. Thus, we are between cases 2 and 3 of the MT and the MT cannot be applied. However, the extensions of the MT (see the Wikipedia page) apply.

(4) The critical value is $c = \log_4(16) = 2$. For any ϵ, $0 < \epsilon < 1, f(n) = n \in O(n^{2-\epsilon})$ and we are in Case 1, so that $T(n) = \Theta(n^2)$.

Problem 2:

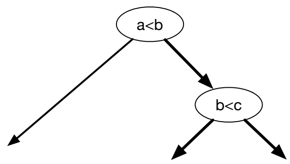
The recurrence obviously has many different solutions, depending on the initial value. By setting $a_0 = 2^0 - 1 = 0$ or $a_1 = 2^1 - 1 = 1$, we ensure that the base case of the induction is true. For the induction step, we assume the Induction Hypothesis that $a_j = 2^j - 1$ for all $j \in \mathbb{N}, j < i$. We need to show that $a_i = 2^i - 1$ as well. We calculate

$$\begin{aligned}
a_i &= 2 \cdot a_{i-1} + 1 && \text{(Recurrence)}\\
&= 2 \cdot (2^{i-1} - 1) + 1 && \text{(Induction Hypothesis)}\\
&= 2^i - 2 + 1 && \text{(Arithmetic)}\\
&= 2^i - 1 && \text{(Arithmetic)}
\end{aligned}$$

which was to be shown.

Problem 3:

(a) If we are given three elements a, b, c, then there are six possibilities for selecting the maximum and the minimum, namely abc; acb, bac, bca, cab, and cba, corresponding to all the permutations of the three elements. As seen in class, if two comparisons were possible to extract both maximum and minimum of the three elements, then we would have an algorithm that orders the three elements in a binary tree with two interior nodes. However, a binary tree with two interior nodes can only have three leaves, so it would not be able to distinguish the six possible cases.



We can easily write an algorithm with three comparisons per runs.

```
def minmax(a,b,c):
     if a<b<c:
          return a,c
     elif a<c<b:
          return a,b
     elif b<a<c:
          return b,c
     elif b<c<a:
          return b,a
     elif c<a<b:
          return c,b
     elif c<b<a:
          return c,a
```

(b) Assume $n \equiv 0 \pmod 3$, so $n = 3m$ for an integer m. This gives us $3m$ comparisons within each group, and then $m-1$ comparisons to find the maximum of the group maxima and $m-1$ comparisons to find the minimum of the group minima. This gives us a total of $5m-2 = \frac{5n}{3} - 2$ comparisons in this case.

Assume now $n \equiv 1 \pmod 3$, so that $n = 3m+1$. We still have $3m$ comparisons within each group, but now have $m+1$ possible maxima and $m+1$ possible minima, so that we have now $5m = \frac{5(n-1)}{3} = \frac{5n}{3} - \frac{5}{3}$ comparisons.

Assume now $n \equiv 2 \pmod 3$, so $n = 3m+2$ for an integer m. There are $3m$ comparisons within the groups and one comparison for the remaining two elements. As before, there are m comparisons to determine the maximum of the group maxima and as well m comparisons to determine the minimum of the group minima. This gives a total of

$$3m + 1 + m + m = 5m + 1 = \frac{5(n-2)+3}{3} = \frac{5n}{3} + \frac{7}{3}$$

comparisons.