Homework 4 Solutions:

Problem 1:

For each of the following recurrences, decide whether the Master Theorem (as in the **book**, not as in Wikipedia) can be applied and if yes, then apply it. Show your work. Identify clearly the parameters a and b and define the function $f(n)$. State whether the MT applies. Define the power of n with which you compare $f(n)$.

(a) $T(n) = 3T(n/2) + n$

Using the MT and its notation, we have $a = 3,\ b = 2,\ f(n) = n$. Thus, $\log_b(a) = \log_2(3)$ ≈ 1.585. With $\epsilon = .5, f(n) = n = O(n^{\log_2(3)-\epsilon})$ and therefore $T(n) = \Theta(n^{\log_2(3)})$.

(b) $T(n) = 3T(n/4) + n^2$

Using the MT and its notation, we have $a = 3,\ b = 4,\ f(n) = n^2$. Thus, $\log_b(a) = \log_4(3)$ ≈ 0.792. With $\epsilon = 0.1$, we have $f(n) = n^2 = \Omega(n^{\log_b(a)+\epsilon})$. We need to evaluate the extra condition: $af(n/b) = 3(n/4)^2 = \dfrac{3n^2}{16} \leq \dfrac{1}{2} \cdot n^2 = \dfrac{1}{2} \cdot f(n)$, thus $T(n) = \Theta(n^2)$.

(c) $T(n) = 4T(n/2) + \log(n)\sqrt{n}$

We have $a = 4$ and $b = 2$ so that because of $\log_4(2) = 1/2$, we have to compare $\sqrt{n}$ with $\sqrt{n} \cdot \log(n)$. As $\lim_{n\to\infty} \dfrac{\sqrt{n}\log(n)}{\sqrt{n}} = \lim_{n\to\infty} \log(n) = \infty$, $\sqrt{n}\log(n) = \Omega(\sqrt{n})$ and so we can only be in Case 3. However, $\lim_{n\to\infty} \dfrac{n^{1/2}\log(n)}{n^{1/2+\epsilon}} = \lim_{n\to\infty} \dfrac{\log(n)}{n^{\epsilon}} =^{\text{LH}} \lim_{n\to\infty} \dfrac{1}{n^{1+\epsilon}} = 0$, for any $\epsilon > 0$, $\sqrt{n}\log n \notin \Omega(n^{1/2+\epsilon})$, so that we are in between Cases 2 and 3. Therefore, the MT does not apply.

(d) $T(n) = \dfrac{2}{3}T(n/2) + \dfrac{1}{3}n$

We have $a = 2/3$. Thus, MT does not apply.

(e) $T(n) = 5T(n/7) + n\cos(n\pi)$

We have $a = 5$ and $b = 7$. However, $n\cos(n\pi)$ is not a positive function, so the MT does not apply.

(f) $T(n) = 4T(\dfrac{n}{16}) + 2^n$

We have $a = 4$ and $b = 16$. Thus, $\log_b(a) = \log_{16} 4 = \frac{1}{2}$. Obviously, $2^n \in \Omega(n^{1/2+1/2})$. The regularity condition becomes $4(f(n/16)) \leq cf(n)$. But the left side evaluates to $4 \cdot 2^{\frac{n}{16}} = 2^{2+\frac{n}{16}}$ which is smaller than 2^n whenever $n > 2$. Thus, $T(n) = \Theta(2^n)$.

(g) $T(n) = 2T(\frac{n}{2}) + 2n\log(n)$

We have $a = 2$ and $b = 2$. As $\log_2(2) = 1$, we compare n with $2n\log(n)$. Because $\lim_{n\to\infty} \frac{2n\log(n)}{n} = \infty$, we have $2n\log(n) \notin \Theta(n)$. However, for $0 < \epsilon < 1$:

$\lim_{n\to\infty} \frac{2n\log(n)}{n^{1+\epsilon}} = \lim_{n\to\infty} \frac{2\log(n)}{n^\epsilon} =^{\text{LH}} \lim_{n\to\infty} \frac{2}{\epsilon n^{\epsilon-1} \cdot n} = \lim_{n\to\infty} \frac{2}{\epsilon n^\epsilon} = 0$, so that $2n\log(n) \notin \Omega(n^{1+\epsilon})$. Thus, we are neither in case 2 nor 3 and the MT does not apply.

Problem 2:

Show that $T(n) = T(n-1) + n + 1$ implies that $T(n) \leq Cn^2$ as long as $C \geq 1$ and $C \geq T(1)$.

We show this by induction. The induction base is already given. For the induction step, we calculate

$T(n+1) = T(n) + (n+1) \leq Cn^2 + n + 1$

Now $C(n+1)^2 = Cn^2 + 2Cn + C > Cn^2 + 2n + 1$, which gives the desired inequality.

Problem 3:

Given the following Python program, prove the loop invariant $\text{acc} = \frac{i(i+1)}{2}$.

```
def litgau(n):
   i = 0
   acc = 0
   while i <= n:
      i += 1
      acc += i
   return acc
```

The loop invariant is true before the while loop starts. Assume it is true before an iteration with value j. Thus $\text{acc} = \frac{j(j+1)}{2}$. After the while loop, $\text{acc} = \frac{j(j+1)}{2} + j + 1$ and the new value of j is $j + 1$. According to the loop invariant, the value of acc should be $\frac{(j+1)(j+2)}{2}$. We calculate $\frac{(j+1)(j+2)}{2} = \frac{j^2+3j+2}{2} = \frac{(j^2+j)+2(j+1)}{2} = \frac{j^2+j}{2} + (j+1)$, which is indeed the new value of acc.

Problem 4:

Given the following C-program, show that the loop invariant $y = 2^i - 1$ is true. Deduce the value of y after the function has run.

```
extern int i;

y=0;
for(i=0; i<=n; i++)    {
    y += pow(2,i);
}
```

At the beginning, y=0, i=0, and $2^i - 1 = 1 - 1 = 0$, so that the loop invariant is true. Before the execution of the loop with a given value of i, we have by assumption $y = 2^i - 1$. During the execution of the loop, y is incremented by 2^i. The new value of y is $2^i - 1 + 2^i = 2^{i+1} - 1$ Then i is also incremented. Therefore, the loop invariant holds.