Homework 2

Nick Alvarez

Q1

a) $T(n) = 7T\left(\frac{n}{2}\right) + n^2$

$a = 7,\ b = 2,\ \log_2 7 = 2.807$

Compare $n^{2.8}$ with $f(n) = n^2 \rightarrow$ case 1

$\rightarrow f(n) = O(n^{2.8-\varepsilon}) \quad \varepsilon = 0.807\ldots\ldots$

$T(n) = \Theta(n^2)$

b) $T(n) = T\left(\frac{n}{9}\right) + n$

$a = 1 \quad b = 9 \quad \log_9 1 = 0.954$

Compare $n^{0.95}$ with $f(n) = n \rightarrow$ case 3

$f(n) = \Omega(n^{0.95+\varepsilon}) \quad \varepsilon = 0.05$

$\hookrightarrow f\left(\frac{n}{b}\right) \leq cf(n)$

$n/9 \leq cn \qquad c = \frac{1}{2} < 1$ ✓

$T(n) = \Theta(n)$

c) $T(n) = T(n-1) + 5$

$\hookrightarrow T(n-1) = T(n-2) + 5$

$T(n) = 5 + [5 + T(n-2)]$

$\hookrightarrow T(n-2) = T(n-3) + 5$

$\vdots$

$\hookrightarrow T(1)$

$T(n) = \Theta(n)$

Q2

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

					pb. size	# prob
level 0:		n			n	1
level 1:	$n/2$		$n/2$		$n/2$	2
level 2:	$n/4$	$n/4$	$n/4$	$n/4$	$n/4$	4
	$\vdots$	$\vdots$	$\vdots$	$\vdots$		
		$n/2^i$			$n/2^i$	2^i
					$\hookrightarrow = 1$	

$n = 2^i \rightarrow \log n - i$

$$T(n) = \sum_{i=0}^{\log n - 1} 2^i \left(\frac{n}{2^i}\right) + 2^{\log n} \cdot T(1)$$

$$= n \sum_{i=0}^{\log n - 1} 1 + n\,T(1)$$

$n \log n$

$T(n) = \Theta(n \log n)$

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

Guess: $O(n^2)$

Prove: $T(n) \leq cn^2$

Assume: $T(k) \leq ck^2 \quad \forall k < n$

$k = n/2 \quad : \quad T(n/2) \quad = c(n/2)^2$

$T(n) = 2T(n/2) + n^2 \leq 2c(n/2)^2 + n^2 =$

$= \cancel{2}c\left(\frac{n^2}{\cancel{4}}\right) + n^2 \leq c/2(\cancel{n^2}) + \cancel{n^2} \leq c\cancel{n^2}$

$c/2 + 1 \leq c$. In this case, $\boxed{c = 2}$ c could equal 1, let's check

Base: $n_0 = 1 \quad T(1) = 2T(0) + 1 = 1 \leq ck^2 = 2(1)$ ✓

$n_0 = 2 \quad T(2) = 2T(1) + 4 = \underset{\leq 1}{T(0)} + \underset{\leq 1}{T(0)} + 4 = 6 \leq ck^2 = 2(4)$

$6 \leq 8$

$\boxed{n_0 = 2, \ c \geq 2} \qquad T(n) = \Theta(n^2)$

Q3

a) $S(n) = 1^3 + 2^3 + \cdots + n^3$

```
if n=1
    return 1;
else
    return S(n-1)+n*n*n;
```

$m(n)$ = # multiplications

$m(1) = 0$

$m(n) = m(n-1) + 2$

$\hookrightarrow m(n-2) + 2 + 2$

$\hookrightarrow m(n-3) + 2 + 2 + 2$

$\vdots$

$\rightarrow m(n-i) + 2i \qquad i = n-1$

$m(n-n+1) + 2(n-1)$

$\boxed{m(n) = m(1) + 2n - 2}$

b) Non recursive algorithm $S(n)$

for $i=2$ to n

$x \mathrel{+}= i \cdot i \cdot i$

$m(n) \sum_{i=2}^{n} 2 = 2n-2$

Q5

$T(n) = T(n^{1/2}) + 1$

$\hookrightarrow T(n^{1/4}) + 1 + 1$

$\hookrightarrow T(n^{1/8}) + 1 + 1 + 1$

$\vdots$

$\hookrightarrow T(n^{1/2^i}) + i$

$n^{1/2^i} = 1 \rightarrow$

Masters: Let $n = 2^m$ and $T(2^m) = S(m)$

We know from attempted iteration above, n's power is being halved each time. m in this case represents the power being halved. So we can say:

$S(m) = S(m/2) + 1$

$\hookrightarrow S(m/4) + 1 + 1$

$\hookrightarrow S(m/8) + 1 + 1 + 1 \ldots$

$\hookrightarrow S(m/2^i) + i$

$2^i = K = \log_2 m \qquad S(m) = S(m/2) + 1$

Case 2: $S(m) = \Theta(m^{\log_b a}) \rightarrow \Theta(m^K)$

$\rightarrow S(m) = \Theta(m^K \log m)$

So $S(m) = \log m$

$n = 2^m$ so n must also be a log function

That leaves us with $\log(\log n)$

So $T(n) = \Theta(\log(\log n))$