Adam Cameron

HW #1

1)

(final)(i) $f(x) = \Theta[g(x)] \rightarrow c_1 g(x) \leq f(x) \leq c_2 g(x)$

$\forall x \in X \rightarrow (0,1)$

$$e^x = \sum_{i=0}^{m-1} \frac{x^i}{i!} + \Theta(x^m) \rightarrow \sum_{i=0}^{\infty} \frac{x^i}{i!} = \sum_{i=0}^{m-1} \frac{x^i}{i!} + \Theta(x^m)$$

$$\Theta(x^m) = \sum_{i=m}^{\infty} \frac{x^i}{i!}$$

(c_1 part)

$$\sum_{i=0}^{m-1} \frac{x^i}{i!} + c_1 x^m \leq \sum_{i=0}^{\infty} \frac{x^i}{i!} \leq c_2 x^m + \sum_{i=0}^{m-1} \frac{x^i}{i!}$$

$$c_1 x^m \leq \sum_{i=m}^{\infty} \frac{x^i}{i!} \leq c_2 x^m$$

$$c_1 \leq \frac{1}{x^m} \sum_{i=m}^{\infty} \frac{x^i}{i!} \leq c_2$$

$$\sum_{i=m}^{\infty} \frac{x^i}{i!} = \frac{x^m}{m!} \sum_{i=m+1}^{\infty} \frac{x^i}{i!}, \quad \frac{1}{x^m}\left(\frac{x^m}{m!} \sum_{i=n+1}^{\infty} \frac{x^i}{i!}\right)$$

$$= \frac{1}{m!} \sum_{m+1}^{\infty} \frac{x^i}{i!}$$

$$C_1 \leq \frac{1}{m!} \sum_{i=m+1}^{\infty} \frac{x^i}{i!} \leq C_2,$$

$$C_1 = \frac{1}{m!};$$

$$\sum_{i=m}^{\infty} \frac{x^i}{i!} \leq C_2 x^m,$$

$$\sum_{i=0}^{\infty} \frac{x^i}{i!} - \sum_{i=0}^{m-1} \frac{x^i}{i!} \leq C_2 x^m,$$

$$e^x - \sum_{i=0}^{m-1} \frac{x^i}{i!} \leq C_2 x^m,$$

$$\frac{1}{x^m}\left(e^x - \sum_{i=0}^{m-1} \frac{x^i}{i!}\right) \leq C_2,$$

$$\frac{e^x}{x^m} - \frac{1}{x^m} \sum_{i=0}^{m-1} \frac{x^i}{i!} \leq C_2,$$

$$x = 1, \quad C_2 = \frac{e^x}{x^m} \rightarrow \frac{e^1}{1^m} = e \quad \therefore$$

(x is always <1)

$$T(n) = aT(n/b) + \Theta(n^k \log_p n) \quad p<0$$

$$T(n) = \Theta(n^k \log_p n)$$

(b)

- (4.1g) $T(n) = 2T(n/4) + \sqrt{n}$
- verify: $a=2,\ b=4,\ f(n)=\sqrt{n},\ k=\frac{1}{2},\ p=0$

$\log_b a = \log_4 2,\ n^{1/2} = n^{\log_4 2 + \varepsilon},\ \log_4 2 = \frac{1}{2},\ \varepsilon = 0$

$T(n) = \Theta(\sqrt{n} \log n)$

- recursion tree:

$\sqrt{n} \rightarrow \sqrt{n}$ $\qquad f\left(\frac{n}{4^L}\right) = 1,\ \frac{n}{4^L} = 1,$

$\sqrt{\frac{n}{4}} \quad \sqrt{\frac{n}{4}} \rightarrow \sqrt{n}$ $\qquad 4^L = n,\ L = \log_4 n$

$\sqrt{\frac{n}{16}} \quad \sqrt{\frac{n}{16}} \quad \sqrt{\frac{n}{16}} \quad \sqrt{n/16} \rightarrow \sqrt{n}$ $\qquad = \frac{\log n}{\log 4}$

...

$1 \ \ldots \qquad 1 \rightarrow \sqrt{n}$

$$T(n) = \sum_{L=0}^{\log_4 n - 1} L\sqrt{n} = \sqrt{n}\left(\frac{\log n}{\log 4}\right),\ T(n) = \Theta(\sqrt{n} \log n)$$

- (4.4j) $T(n) = \sqrt{n}\,T(\sqrt{n}) + n$

$= \sqrt{n}\,T(n/\sqrt{n}) + n$

~~verify:~~

verify: $f(n) = n,\ \log_b a = \log_{\sqrt{n}} \sqrt{n} = \underline{1}$

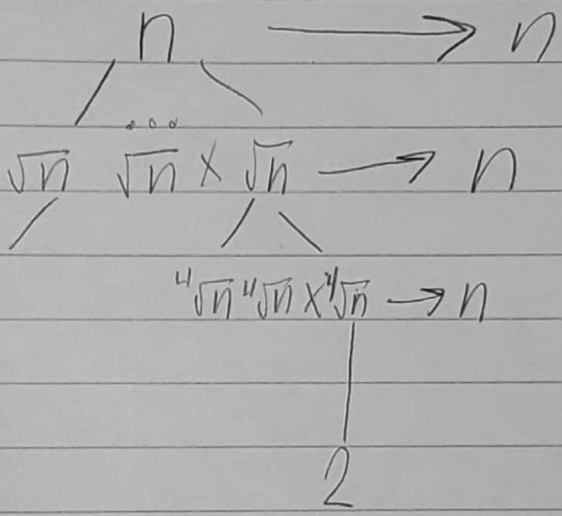
$a = \sqrt{n},\ b = \sqrt{n},\ k = 1,\ p = 0$

Final Draft

HW #1

(4.4:) recursion tree

$T(n) = \sqrt{n}\, T(\sqrt{n}) + n$



$T(2) = n^{(\frac{1}{2})^L} = 2$, $(\frac{1}{2})^L = \log_2 n$;

$(\frac{1}{2})^L = x$, $L = \log_{\frac{1}{2}} x = \log_{\frac{1}{2}}(\log_2 n) =$

$$\frac{\log(\log_2 n)}{\log \frac{1}{2}} = \frac{\log(\log n / \log 2)}{\log \frac{1}{2}}$$

$$T(n) = n\left(\frac{\log(\log n / \log 2)}{\log \frac{1}{2}}\right) = \Theta(n \log(\log n))$$

verify (master thm): • $T(n) = \sqrt{n}\, T(n/\sqrt{n}) + n$,
$a = \sqrt{n}$, $b = \sqrt{n}$

$n^{\log_{\sqrt{n}} \sqrt{n}} = n^1 = n$, $\Theta(n^{\log_{\sqrt{n}} \sqrt{n}}) = \Theta(n) = f(n) = n$

therefore $T(n) = \Theta(n' \log(\log n))$

$\boxed{\text{final draft}}$

HW #1)

Problem 1

1)

(C) (4.1h) $T(n) = T(n-2) + n^2$

Base: $T(0) = 0$ $(0^2 = 0)$

$$T(n) = T(n-2) + n^2$$
$$= T(n-4) + (n-2)^2 + n^2$$
$$= T(n-6) + (n-4)^2 + (n-2)^2 + n^2$$
$$= T(n-k) + (n-2(k-1))^2 + (n-2(k-2))^2 + \dots + n^2$$

$$\left(\begin{array}{ll} T(n-k) = T(0) = 0 & n - 2(\tfrac{n}{2}) = n - n = 0 \\ n - k = 0 & \\ n = k & \end{array} \right)$$

$$= \sum_{i=0}^{n/2} (n-2i)^2 = \sum_{i=0}^{n/2} (n^2 - 4ni + 4i)$$

$$= \Theta\left(\frac{n^3}{2}\right) + \Theta(2n^2) + (2n)$$

$T(0) = T(n-k) = 0$

$= \Theta(n^3) + \Theta(n^2) + \Theta(n)$

$= \Theta(n^3)$

4.4h $T(n) = T(n-1) + \log n$

Base Case: $T(0) = 1$

$T(n) = T(n-2) + \log(n-1) + \log n$

$= T(n-3) + \log(n-2) + \log(n-1) + \log n$

$= T(n-k) + \log(1) + \log(2) + \dots + \log n$

$\left(\begin{array}{l} T(n-k) = T(0) = 1 \\ n-k = 0 \\ n = k \end{array} \right)$

$T(n) = T(n-k) + \log 1 + \log 2 + \dots + \log(n-1) + \log n$

$= T(0) + \log n!$

$= 1 + \log n! = \Theta(1) + \Theta(n \log n)$

$= \Theta(n \log n)$

Final Draft

HW 2

Problem 2

1)

Insertion Sort:

		T(n)
C1	for i=0 to n	n
C2	key = A[i]	$n-1$
e3	j = i-1	$n-1$
C4	while j>=0 and A[j] > key	$\sum_{j=1}^{n-1} t_j$
C5	A[j+1] = A[j]	$\sum_{j=1}^{n-1} t_j - 1$
C6	j = j-1	
C7	end while	
C8	A[j+1] = key	$n-1$
C9	end for	

$$E[T(n)] = C_1 n + (C_2 + C_3 + C_3)(n-1) + C_4 \sum_{j=1}^{n-1} E(t_j) + (C_5 + C_6) \sum_{j=1}^{n-1} (E(t_j) - 1)$$

best j = 1 (already sorted), worst j = j, average (j-1)/2

$$T(n) = C_1 n + (C_2 + C_3 + C_8)(n-1) + C_4 \sum_{j=1}^{n-1} \frac{(j-1)}{2}$$
$$] + (C_5 + C_6)((n-1)$$
$$+ (C_5 + C_6) \sum_{j=1}^{n-1} \frac{(j-1)}{2} - 1$$

$= C_1 n + (C_2 + C_3 + C_8)(n-1) + C_4 (n-1)(n-1)/2$

$+ (C_5 + C_6)(n-1)(n-3)/2$

$= O(n) + O(n-1) + O((n-1)^2/2) + O((n-1)(n-3)/2)$

$= O(n) + O(n^2)$

$= \underline{O(n^2)}$

Max Element :

```
def  Max Element (array)                      T(n)
C1 | largest = array[0]                     |  1
C2 | for i from 1 to length (array)         |  n
C3 |     if largest < array[i]              |  n-1
C4 |         largest = array[i]             |  n-1
C5 | return largest                         |  1
```

$E[T(n)] = C_1 * 1 + C_2 * n + (C_3 + C_4)(n-1) + C_5 * 1$

$= (C_1 + C_2) + C_2 n + (C_3 + C_4)(n-1)$

$= O(1) + O(n) + O(n-1)$

$= \underline{O(n)}$

Final Draft

HW1

Problem 2

1)

$\boxed{\text{Solve w/ notes}}$

Merge Sort:

```
                                                     T(n)
def MergeSortSetup(vector<int> input)
C1   left = 0                                        1
C2   right = input.size - 1                          1
C3   Call MergeSort (input, left, right)             1

def MergeSort(vector<int> input, left, right)
     If right > left
find midpoint
C4   middle = left + (right - left)/2                1
C5   Call MergeSort (input, left, middle)            T(n/2)
C6   Call MergeSort (input, middle+1, right)         T(n/2)
C7   Call Merge (input, left, middle, right)         f(n)

def Merge (input, left, middle, right)
C8   left_length = middle - left + 1                 1
C9   right_length = right - middle                   1
C10  L_array[left_length]                            1
C11  for i from 0 to n:                              n
C12       L_array = input[left+i]                    n-1
```

C13	R_array [right_length]	1
C14	for i from 0 to right_length	n
C15	R_array = input [middle + i + 1]	n-1
C16	l_ptr = 0	1
C17	r_ptr = 0	1
C18	index = left	1
C19	while (l_ptr < left_length and r_ptr < right_length)	n
C20	if (L_array [l_ptr] < R_array [r_ptr])	n-1
C21	input [index] = L_array [l_ptr]	n-1
C22	index ++	n-1
C23	l_ptr ++	n-1
C24	else	n-1
C25	input [index] = R_array [r_ptr]	n-1
C26	index ++	n-1
C27	r_ptr ++	n-1
C28	while (l_ptr < left_length)	n
C29	input [index] = L_array [l_ptr]	n-1
C30	index ++	n-1
C31	l_ptr ++	n-1
C32	while (r_ptr < right_length)	n
C33	input [index] = R_array [r_ptr]	n-1
C34	index ++	n-1
C35	r_ptr ++	n-1
C36		

HW #1

Problem 2

Merge Sort Cont)

$$E(f(n)) = 1(C_8 + C_9 + C_{10} + C_{13} + C_{16} + C_{17} + C_{18}) + n(C_{11} + C_{14} + C_{19} + C_{28} + C_{32}) + (n-1)(C_{12} + C_{15} + C_{20} + C_{21} + C_{22} + C_{23} + C_{24} + C_{25} + C_{26} + C_{27} + C_{29} + C_{30} + C_{31} + C_{33} + C_{34} + C_{35})$$

$$= \Theta(1) + \Theta(n) + \Theta(n-1)$$

$$= \Theta(n)$$

$$E(T(n)) = C_1(1) + (C_2 + C_3)(T(n/2)) + C_7(f(n))$$

$$= \Theta(1) + 2(T(n/2)) + \Theta(n)$$

$$= 2T(n/2) + \Theta(n);$$

Master's Theorem

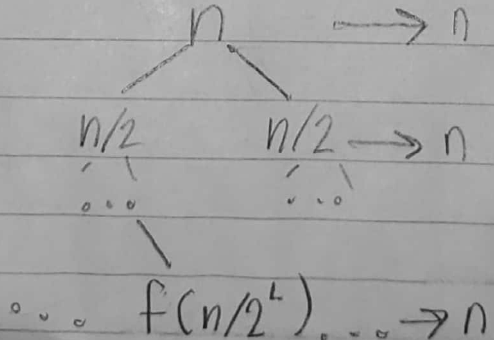
$$T(n) = aT(n/b) + f(n),$$

$$a = 2,\ b = 2,\ f(n) = n,\ \log_b a = \log_2 2 = 1,$$

$$f(n) = n^1 = n^{\log_2 2}$$

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

Recursion Tree



final draft

HW # 1
Problem 2

Merge Sort (cont)

$n/2^L = 1,$

$n = 2^L, \; L = \log_2 n = \dfrac{\log n}{\log 2}$

$T(n) = n(\log n / \log_2)$

$= n(\log n)/\log 2$

$= O(n \log n)$