CS3230 HW 2
TAN SOON JIN
A0112213E
Tut 3

S1

g6(n) << g4(n) << g8(n) << g2(n) << g5(n) << g3(n) << g7(n) << g1(n)

S2

a)

$T(n) = 4T(n/2) + \Theta(n^2 \lg^3 n)$
$a = 4$, $b = 2$, $n \Rightarrow n^2$
$f(n) = \Theta(n^2 \lg^3 n)$ where $k = 3$

$T(n) = \Theta(n^2 \lg^4 n)$

b)

$T(n) = 4T(n/2) + 5n$

Using recursion tree method where $c = 5$

		cn			cn
$c(n/2)$	$c(n/2)$		$c(n/2)$	$c(n/2)$	$2cn$
$c(n/4)$ $c(n/4)$ $c(n/4)$ $c(n/4)$					$4cn$
					$8cn$

Summation is a geometric series with $r = 2$, k = height of tree

$= cn(1+2+4+8+16 \ldots\ldots\ldots\ldots 2^k)$

$= cn(2^{\log(n)} - 1)$

$= cn(n^{\log(2)})$

$= cn^2$

$= \Theta(n^2)$

c)

$U(n) = [\ 2/(n-1) \quad \Sigma\ U(k)\ ] + 5n$

$(n-1)U(n) = 2\ \Sigma\ U(k) + (5n^2 – 5n)$ …... Equation 1

$(n-2)U(n-1) = 2\ \Sigma U(k) + (5n^2 – 15n+10)$ ….... Equation 2

1 – 2

$(n – 1)U(n) = n\ U(n-1) + 10(n – 1)$......... Divide by n(n-1)

$U(n)/n = U(n-1)/(n-1) + 10/n$

Telescoping the recurrence relation

$U(n-1)/(n-1) = U(n-2)/(n-2) + 10/(n-1)$
$U(n-2)/(n-2) = U(n-3)/(n-3) + 10/(n-2)$
.
.
.
.
$= U(1)/1 + 10 * \Sigma\ 1/k$

$U(n) = (n-1)*10*H(n) + \Theta(1)$
$= \Theta(n \log n)$

S3

Initial State

A = [3,2,1,7,1,7,2,5,1,5,7,4]

C = [0,0,0,0,0,0,0]

I = 3 , C = [0,0,1,0,0,0,0]
I = 2 , C = [0,1,1,0,0,0,0]
I = 1 , C = [1,1,1,0,0,0,0]
I = 7 , C = [1,1,1,0,0,0,1]
I = 1 , C = [2,1,1,0,0,0,1]
I = 7 , C = [2,1,1,0,0,0,2]
I = 2 , C = [2,2,1,0,0,0,2]
I = 5 , C = [2,2,1,0,1,0,2]
I = 1 , C = [3,2,1,0,1,0,2]
I = 5 , C = [3,2,1,0,2,0,2]
I = 7 , C = [3,2,1,0,2,0,3]
I = 4 , C = [3,2,1,1,2,0,3]

C" = [3,5,6,7,9,9,12]

Going through A

C" = [3,5,5,7,9,9,12] B = [0,0,0,0,0,3,0,0,0,0,0,0]

C” = [3,4,5,7,9,9,12] B = [0,0,0,0,2,3,0,0,0,0,0,0]

C” = [2,4,5,7,9,9,12] B = [0,0,1,0,2,3,0,0,0,0,0,0]

C” = [2,4,5,7,9,9,11] B = [0,0,1,0,2,3,0,0,0,0,0,7]

C” = [1,4,5,7,9,9,11] B = [0,1,1,0,2,3,0,0,0,0,0,7]

C” = [1,4,5,7,9,9,10] B = [0,1,1,0,2,3,0,0,0,0,7,7]

C” = [1,3,5,7,9,9,10] B = [0,1,1,2,2,3,0,0,0,0,7,7]

C” = [1,3,5,7,8,9,10] B = [0,1,1,2,2,3,0,0,5,0,7,7]

C” = [0,3,5,7,8,9,10] B = [1,1,1,2,2,3,0,0,5,0,7,7]

C” = [0,3,5,7,7,9,10] B = [1,1,1,2,2,3,0,5,5,0,7,7]

C” = [0,3,5,7,7,9,9] B = [1,1,1,2,2,3,0,5,5,7,7,7]

C” = [0,3,5,6,7,9,9] B = [1,1,1,2,2,3,4,5,5,7,7,7]

S4

a)

Description:

Create an array B[1....n]. Scan through A where A[i] = x. Set B[x] to 1 during the scan.
Scan through B to look for 0 cell.

Correctness

Since there is only one missing integer in the array, a full linear scan is bound to find the missing integer.

Runtime

$f(n) = \Theta(n)$ - linear scan

Termination:

Since the array has finite elements, each iteration the counter is decremented and eventually it will hit the upper bound of the loop and terminate. For scanning of array B, it will terminate when 0 is found or the counter reached the upper bound

b)

Description:

It is similar to binary search by partitioning elements according to their kth bit

1. Create two arrays A and B of size n/2 and S, one empty string of length k.
2. Do bit query on kth bit for N numbers.
3. store numbers with 1 from the query in A. store numbers with 0 from query in B..

4. chosen array = min(length(A), length(B)).
   if chosen is A append 1 to S, else append 0 to S.
5. Repeat step 5 until selected array is empty
6. Finally, S is the binary representation of your missing integer

Runtime analysis

Can be represented as recurrence relation $T(n) = T(n/2) + \Theta(n)$ using Master's Theorem
$T(n) = \Theta(n)$

Termination & Correctness

As the required number of queries is halved every iteration, it will terminate when one of the array is empty.

Example

1. First bit query on N numbers on kth bit

A = [4,6,7]   B = [1,2,3,5]

chosen array = A since min(length(A), length(B))
S = '1'

2. Bit query on chosen array [4,6,7] on k -1 bit

A = [4,7]   B = [6]

chosen array = B
S = '10'

3. Bit query on chosen array [6] on k -2 bit

A = []   B = [6]

chosen array = A
S = '101'
terminates since A is empty

Result = 101 = 5 is the missing integer