BLG 335E ANALYSIS OF ALGORITHMS I
MIDTERM - NOVEMBER 13, 2013, 13:30-15:30 PM (2 hours)

1 (10 pt)	2 (18 pt)	2 (22 pt)	3 (15 pt)	4 (30 pt)	5 (15 pt)	Total (100 pt)

On my honor, I declare that I neither give nor receive any unauthorized help on this exam.

Student Signature:___________________________

Write your name on each sheet.
Write your answers neatly (in English) in the space provided for them.
You must show all your work for credit.
Books and notes are closed.
Good Luck!

Q1[10 points]:

1a) Is $2^{n+1} = O(2^n)$?

1b) Is $2^{2n} = O(2^n)$?

Show your work. Define c, n_0.

Hint: Definition if O-notation (page 44 from textbook)

Q2[18 points]:

Find the solutions for the following recurrences. Feel free to use one of the three methods: substitution method, recursion-tree method, master method. **Show your work.**

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

b) T(n) = T(n/2) + T(n/4) + T(n/8) + n

c) T(n) = T(n-1) + lgn

HINT: If you want to benefit from **MASTER THEOREM** for Q2:

Master theorem:
Let a≥1 and b>1 be constants, let f(n) be a function, and let T(n) be defined on the nonnegative integers by the recurrence

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

where we interpret n/b to mean either $\lfloor n/b \rfloor$ or $\lceil n/b \rceil$. Then *T(n)* can be bounded asymptotically as follows.

1. If $f(n) = O(n^{\log_b a-\epsilon})$ for some constant ϵ>0, then $T(n) = \theta(n^{\log_b a})$.
2. If $f(n) = \theta(n^{\log_b a})$, then $T(n) = \theta(n^{\log_b a} \lg n)$.
3. If $f(n) = \Omega(n^{\log_b a+\epsilon})$ for some constant ϵ>0, and if $af(n/b) \le cf(n)$ for some constant *c<1* and all sufficiently large n, then $T(n) = \theta(f(n))$.

Q3) [22 pts]

3a) (7 pts) Given a sample space *S* and an event *A* in the sample space *S*, let X_A = I {A}. Show that E [X_A] = Pr {A}. **Note that I {A} is indicator random variable.**

3b) (15 pts) How many people do you need in a room to have at least 2 with the same birthday? Assume that birthdays are distributed equally among all days of the year and neglect leap years, that is you can take 1 year = 365 days)
Hint. Use indicator random variable

PART-A SOLUTIONS

1)

$2^{n+1} = O(2^n)$, but $2^{2n} \neq O(2^n)$.

To show that $2^{n+1} = O(2^n)$, we must find constants $c, n_0 > 0$ such that

$0 \leq 2^{n+1} \leq c \cdot 2^n$ for all $n \geq n_0$.

Since $2^{n+1} = 2 \cdot 2^n$ for all n, we can satisfy the definition with $c = 2$ and $n_0 = 1$.

To show that $2^{2n} \neq O(2^n)$, assume there exist constants $c, n_0 > 0$ such that

$0 \leq 2^{2n} \leq c \cdot 2^n$ for all $n \geq n_0$.

Then $2^{2n} = 2^n \cdot 2^n \leq c \cdot 2^n \Rightarrow 2^n \leq c$. But no constant is greater than all 2^n, and so the assumption leads to a contradiction.

2)

a)

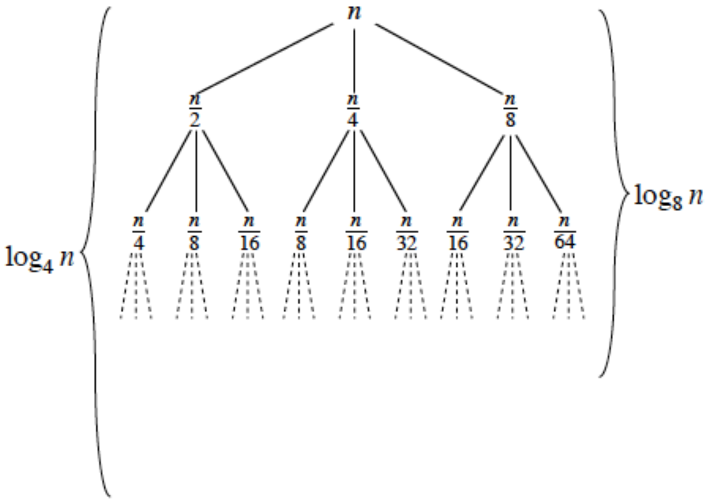
$T(n) = 3T(n/2) + n \lg n$

We have $f(n) = n \lg n$ and $n^{\log_b a} = n^{\lg 3} \approx n^{1.585}$. Since $n \lg n = O(n^{\lg 3 - \epsilon})$ for any $0 < \epsilon \leq 0.58$, by case 1 of the master theorem, we have $T(n) = \Theta(n^{\lg 3})$.

b)

$T(n) = T(n/2) + T(n/4) + T(n/8) + n$

Using the recursion tree shown below, we get a guess of $T(n) = \Theta(n)$.



$$n$$

$$n(\tfrac{4+2+1}{8}) = \tfrac{7}{8}n$$

$$\begin{aligned} & n(\tfrac{1}{4}+\tfrac{2}{8}+\tfrac{3}{16}+\tfrac{2}{32}+\tfrac{1}{64}) \\ &= n\tfrac{16+16+12+4+1}{64} \\ &= n\tfrac{49}{64} = \tfrac{7}{8}^2 n \end{aligned}$$

$$\vdots$$

$$\sum_{i=1}^{\log n} \left(\frac{7}{8}\right)^i n = \Theta(n)$$

We use the substitution method to prove that $T(n) = O(n)$. Our inductive hypothesis is that $T(n) \le cn$ for some constant $c > 0$. We have

$$\begin{aligned} T(n) &= T(n/2) + T(n/4) + T(n/8) + n \\ &\le cn/2 + cn/4 + cn/8 + n \\ &= 7cn/8 + n \\ &= (1 + 7c/8)n \\ &\le cn \qquad \text{if } c \ge 8 . \end{aligned}$$

Therefore, $T(n) = O(n)$.

Showing that $T(n) = \Omega(n)$ is easy:

$$T(n) = T(n/2) + T(n/4) + T(n/8) + n \ge n .$$

Since $T(n) = O(n)$ and $T(n) = \Omega(n)$, we have that $T(n) = \Theta(n)$.

c)

$$T(n) = T(n-1) + \lg n$$

We guess that $T(n) = \Theta(n \lg n)$. To prove the upper bound, we will show that $T(n) = O(n \lg n)$. Our inductive hypothesis is that $T(n) \le cn \lg n$ for some constant c. We have

$$
\begin{aligned}
T(n) &= T(n-1) + \lg n \\
&\le c(n-1)\lg(n-1) + \lg n \\
&= cn\lg(n-1) - c\lg(n-1) + \lg n \\
&\le cn\lg(n-1) - c\lg(n/2) + \lg n \\
&\qquad\qquad (\text{since } \lg(n-1) \ge \lg(n/2) \text{ for } n \ge 2) \\
&= cn\lg(n-1) - c\lg n + c + \lg n \\
&< cn\lg n - c\lg n + c + \lg n \\
&\le cn\lg n \, ,
\end{aligned}
$$

if $-c\lg n + c + \lg n \le 0$. Equivalently,

$$
\begin{aligned}
-c\lg n + c + \lg n &\le 0 \\
c &\le (c-1)\lg n \\
\lg n &\ge c/(c-1) \, .
\end{aligned}
$$

This works for $c = 2$ and all $n \ge 4$.

To prove the lower bound, we will show that $T(n) = \Omega(n\lg n)$. Our inductive hypothesis is that $T(n) \ge cn\lg n + dn$ for constants c and d. We have

$$
\begin{aligned}
T(n) &= T(n-1) + \lg n \\
&\ge c(n-1)\lg(n-1) + d(n-1) + \lg n \\
&= cn\lg(n-1) - c\lg(n-1) + dn - d + \lg n \\
&\ge cn\lg(n/2) - c\lg(n-1) + dn - d + \lg n \\
&\qquad\quad (\text{since } \lg(n-1) \ge \lg(n/2) \text{ for } n \ge 2) \\
&= cn\lg n - cn - c\lg(n-1) + dn - d + \lg n \\
&\ge cn\lg n \, ,
\end{aligned}
$$

if $-cn - c\lg(n-1) + dn - d + \lg n \ge 0$. Since

$$
\begin{aligned}
&-cn - c\lg(n-1) + dn - d + \lg n > \\
&\qquad -cn - c\lg(n-1) + dn - d + \lg(n-1) \, ,
\end{aligned}
$$

it suffices to find conditions in which $-cn - c\lg(n-1) + dn - d + \lg(n-1) \ge 0$. Equivalently,

$$
\begin{aligned}
-cn - c\lg(n-1) + dn - d + \lg(n-1) &\ge 0 \\
(d-c)n &\ge (c-1)\lg(n-1) + d \, .
\end{aligned}
$$

This works for $c = 1$, $d = 2$, and all $n \ge 2$.

Since $T(n) = O(n\lg n)$ and $T(n) = \Omega(n\lg n)$, we conclude that $T(n) = \Theta(n\lg n)$.

3)

a)

Proof Letting $\overline{A}$ be the complement of A, we have

$$\begin{aligned} \mathrm{E}[X_A] &= \mathrm{E}[\mathrm{I}\{A\}] \\ &= 1 \cdot \Pr\{A\} + 0 \cdot \Pr\{\overline{A}\} \quad \text{(definition of expected value)} \\ &= \Pr\{A\} . \end{aligned}$$

b)

We can use indicator random variables to provide a simpler but approximate analysis of the birthday paradox. For each pair (i, j) of the k people in the room, we define the indicator random variable X_{ij}, for $1 \le i < j \le k$, by

$$\begin{aligned} X_{ij} &= \mathrm{I}\{\text{person } i \text{ and person } j \text{ have the same birthday}\} \\ &= \begin{cases} 1 & \text{if person } i \text{ and person } j \text{ have the same birthday}, \\ 0 & \text{otherwise}. \end{cases} \end{aligned}$$

By equation (5.7), the probability that two people have matching birthdays is $1/n$, and thus by Lemma 5.1, we have

$$\begin{aligned} \mathrm{E}[X_{ij}] &= \Pr\{\text{person } i \text{ and person } j \text{ have the same birthday}\} \\ &= 1/n . \end{aligned}$$

Letting X be the random variable that counts the number of pairs of individuals having the same birthday, we have

$$X = \sum_{i=1}^{k} \sum_{j=i+1}^{k} X_{ij} .$$

Taking expectations of both sides and applying linearity of expectation, we obtain

$$\begin{aligned} \mathrm{E}[X] &= \mathrm{E}\left[\sum_{i=1}^{k} \sum_{j=i+1}^{k} X_{ij}\right] \\ &= \sum_{i=1}^{k} \sum_{j=i+1}^{k} \mathrm{E}[X_{ij}] \\ &= \binom{k}{2} \frac{1}{n} \\ &= \frac{k(k-1)}{2n} . \end{aligned}$$

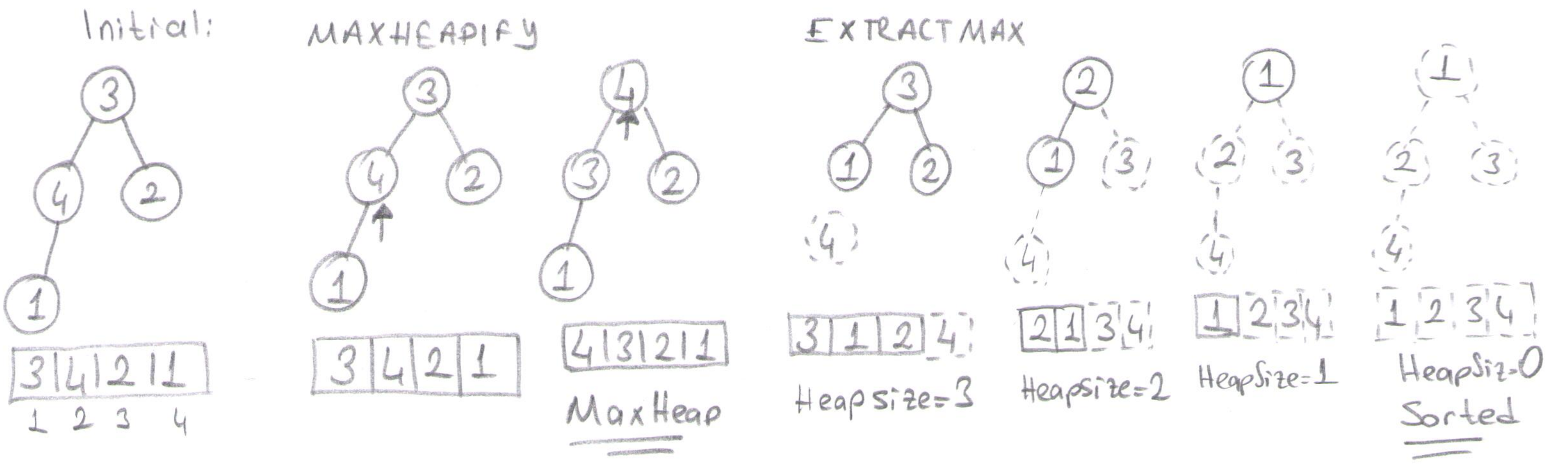
When $k(k-1) \ge 2n$, therefore, the expected number of pairs of people with the same birthday is at least 1. Thus, if we have at least $\sqrt{2n}+1$ individuals in a room, we can expect at least two to have the same birthday. For $n = 365$, if $k = 28$, the expected number of pairs with the same birthday is $(28 \cdot 27)/(2 \cdot 365) \approx 1.0356$.

BLG 335E ANALYSIS OF ALGORITHMS I
MIDTERM - NOVEMBER 5, 2014, 13:30-15:30 PM (2 hours)

1 (10 pt)	2 (18 pt)	3 (22 pt)	4 (20 pt)	5 (12 pt)	6 (18 pt)	Total (100 pt)

Q4) [20points]
4a) [10pts] Sort the array A={3,4,2,1} in increasing (ascending) order using Heapsort. Show all the steps of your work. Use tree representation for the heap.

Which type of Heap do you need to use? Max- heap



4b) [10points]
Sort the array A={3,14,13,11}, in increasing (ascending) order using Radix sort. Show all the steps of your work.

03	03	11	03
14	14	03	11
13	13	13	13
11	11	14	14
	stable sort digit 1	stable sort digit 2	Sorted

Q5) [12 points]

Given the following algorithm which computes the minimum of an array, prove its correctness using a loop invariant.

```
MINIMUM(A)
1 min ← A[1]
2 for i ← 2 to length[A]
3 do if min > A[i]
4     then min ← A[i]
5 return min
```

loop

State the Loop Invariant:

At the begining of the for loop, (on line 2)
min is the minimum of $A[1 \ldots i-1]$

Initialization:

$i = 2$, $min = A[1]$ // line 1

$A[1 .. i-1] = A[1]$

Therefore min is $A[1]$ which is the only minimum array element.

Maintenance: Assume loop invariant true for i, show it true for $i+1$

$\boxed{\text{min is the minimum of } A[1..i-1]}$

(line3) Two cases to consider

if $min > A[i]$

then $A[i]$ is the smallest of $A[1 \ldots i]$

min is assigned to $A[i]$ on line 4

if $min \leq A[i]$

then $A[i]$ is not minimum and min does not need to be changed.

min contains the minimum of $A[1 .. i]$

Therefore at iteration $i+1$, Loop invariant is true.

Termination:

At termination

$i = length[A] + 1$

loop invariant:

min is the minimum of $A[1 \ldots length[A]]$

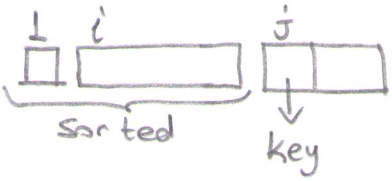
Therefore the algorithm computes the correct value.

Q6) [18pts]: Fill in the following table according to the implementations we learned in class:

Hint: All comparisons used in Insertion, Heap, Merge sort are $\leq$ or $>$

	Worst Case Time Complexity T(n)=	In Place? (Yes or No?)	Stable? (Yes or No?)
Insertion Sort	O(n^2)	Yes	Yes
Heapsort	O($n \log n$)	Yes	No
Mergesort	O($n \log n$)	No	Yes

Insertion Sort



Only if an element is $>$ key it is moved

$i = j - 1$

while $A[i] >$ key move ith element to it

Let $c_a = c_b$ (a & b used to indicate which element)

if $A[j] = c_a$ and $A[j+k] = c_b$

then when sorted the order will be

$c_a \, c_b$

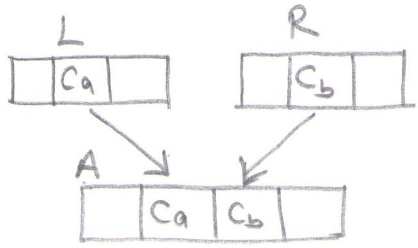
Merge Sort

Merge compares $L[i] \leq R[j]$

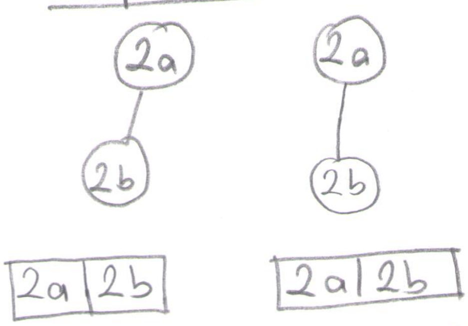
if true moves $L[i]$ to $A[k]$

else moves $R[j]$ to $A[k]$

$c_a \, c_b$ remain in the same order.



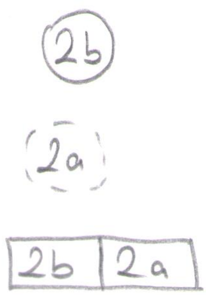
Heap Sort



MaxHeapify: No swaps

if $A[l]$ or $A[r]$ is $> A[i]$

then swap



Extract max swaps 2a & 2b