CPSC 320 Midterm 1

June 29, 2007

Name: __________________________ Student ID: ______________________

Signature: _______________________

- You have 1 hour to write the 5 questions on this examination. A total of 23 marks are available.
- *Justify all of your answers.*
- Keep your answers short. If you run out of space for a question, you have written too much.
- The number in the square brackets next to the question number is the # of marks allocated for that question. Use these to help determine how much time you should spend on each question.
- Use the back of the pages for your rough work.
- *Good luck!*

Question	Marks	Score
1	6	
2	2	
3	4	
4	4	
5	4	
Total	20	

UNIVERSITY REGULATIONS:

- Each candidate should be prepared to produce, upon request, his/her library card.
- No candidate shall be permitted to enter the examination room after the expiration of one half hour or to leave during the first half hour of the examination.
- CAUTION: Candidates guilty of any of the following, or similar, dishonest practices shall be immediately dismissed from the examination and shall be liable to disciplinary action.
  1. Having at the place of writing, or making use of, any books, papers or memoranda, electronic equipment, or other memory aid or communication devices, other than those authorized by the examiners.
  2. Speaking or communicating with other candidates.
  3. Purposely exposing written papers to the view of other candidates. The plea of accident or forgetfulness shall not be received.
- Candidates must not destroy or mutilate any examination material; must hand in all examination papers; and must not take any examination material from the examination room without the permission of a invigilator.

1. [6] Use the master theorem to obtain tight asymptotic bounds for the following recurrences. Please assume that $T(n) \in \Theta(1)$ for small n and that $T(n)$ is increasing. If you cannot use the master theorem to solve the recurrence, example why.

Master Theorem: Let $a \geq 1$ 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.

- If $f(n) \in O\left(n^{\log_b a - \epsilon}\right)$ for some constant $\epsilon > 0$, then $T(n) \in \Theta\left(n^{\log_b a}\right)$.
- If $f(n) \in \Theta\left(n^{\log_b a}\right)$, then $T(n) \in \Theta\left(n^{\log_b a} \log n\right)$.
- If $f(n) \in \Omega\left(n^{\log_b a + \epsilon}\right)$ for some constant $\epsilon > 0$, and if $af(n/b) \leq \delta f(n)$ for some constant $\delta < 1$ and all sufficiently large n, then $T(n) = \Theta(f(n))$.

(a) $T(n) = 2T(\lceil n/2 \rceil) + n + \log n - 2$

$n + \log n - 2 \in \Theta(n)$ because $\log n$ and -2 are low order terms. Now $n^{\log_2 2} = n$, so case 2 of the master theorem applies. Therefore, $T(n) \in \Theta(n \log n)$.

(b) $T(n) = 9(\lfloor n/4 \rfloor) + 5n \log n$

Let $\epsilon = \log_4 9 - \log_4 8$. Now $\log_4 8 = \log_4(2 \cdot 4) = \log_4 2 + \log_4 4 = 3/2$. So $n^{\log_4 9 - \epsilon} = n^{3/2}$. Now $\log n \in O(\sqrt{n})$, so $n \log n \in O\left(n^{3/2}\right) = O\left(n^{\log_4 9 - \epsilon}\right)$. So case 1 holds, and $T(n) \in \Theta\left(n^{\log_4 9}\right)$.

(c) $T(n) = 3T(\lceil n/2 \rceil) + n^2$

$\log_2 3 < \log_2 4 = 2$, so let $\epsilon = 2 - \log_2 3$. Then $n^2 \in \Omega\left(n^{\log_2 3+\epsilon}\right)$. To apply case 3, we must find the appropriate δ.

$$\begin{aligned} 3(n/2)^2 &\leq \delta n^2 \\ 3n^2/4 &\leq \delta n^2 \\ 3/4 &\leq \delta \end{aligned}$$

So $\delta = 3/4$ suffices. Therefore, case 3 holds and $T(n) = \Theta\left(n^2\right)$

2. [2] You are given an unsorted array of n integers that are in the range of $\left[0, n^3 - 1\right]$. Explain how to sort the array in just $O(n)$ time and space.

If we consider each number in base n, then each number has at most 3 digits because $\log_n\left(n^3\right) = 3\log_n n = 3$. Applying radix sort to this representation usings three bucket sorts, which take $O(n)$ time each. Therefore, radix sort runs in $O(n)$ time.

3. [4] Consider the following algorithm.

```
Algorithm Loopy(n)
   for i ← 1 to n do
      j ← i
      while (j ≥ 3) do
         print "Goofy loopy!"
         j ← j/3
```

Use the guess-and-test method to show that running time of this algorithm is $O(n \log n)$. That is, *you must use induction!*

The "while" loop executes $\lfloor \log_3 i \rfloor$ times, so the runtime of this algorithm is

$$T(n) = \sum_{i=1}^{n} \lfloor \log_3 i \rfloor \leq \sum_{i=1}^{n} \log_3 i$$

We prove inductively that $T(n) \leq cn \log_3 n$. Suppose that it holds for $T(n-1)$. Then

$$\begin{aligned} T(n) &= T(n-1) + \lfloor \log_3 n \rfloor \\ &\leq c(n-1)\log_3(n-1) + \lfloor \log_3 n \rfloor \\ &\leq c(n-1)\log_3 n + \lfloor \log_3 n \rfloor \\ &\leq c(n-1)\log_3 n + \log_3 n \\ &\leq cn\log_3 n - c\log_3 n + \log_3 n \\ &\leq cn\log_3 n, \text{ if } c \geq 1 \end{aligned}$$

So inductive step holds regardless of n. We need a base case and n_0. Try $n = 1$.

$$T(1) = 0 \leq c(1)\log(1) = c \times 0$$

So $T(n) \leq n \log n$, for all $n \geq 1$. Hence $T(n) \in O(n \log n)$.

4. [4] You are given an sorted array of n comparable elements, and you want to find the location of a given element k in the array. Prove that **any** algorithm using comparisons must make $\Omega(\log n)$ comparisons in the worst case.

k could be any of the n array elements, so a decision tree must have n leaves. Let h be the height of the decision tree, then the number leaves is less than 2^h. Hence $2^h \geq n$. Taking the base 2 logarthm of both sides, $h \geq \log_2 n$. The height of the tree corresponds to the number of comparisons necessary in the worst case. Hence, the worst-case number of comparisons is $\Omega(\log n)$.

5. [4] Suppose that you are given a $\Theta(n)$ time algorithm to find the median of an array of n elements. Explain how you would find the ith order statistic in $\Theta(n)$ time using this algorithm as a subroutine.

Use the median finding algorithm to find the pivot. Then using that in the partitioning stage of the divide-and-conquer selection algorithm described in class. The resulting recurrence is

$$T(n) = T(\lfloor n/2 \rfloor) + \Theta(n)$$

By case 1 of the master theorem, $T(n) \in \Theta(n)$.