CPSC 320 Midterm 1

June 30, 2006

Name: ________________________ Student ID: ____________________

Signature: _____________________

- You have 1 hour to write the 5 questions on this examination. A total of 30 marks are available.
- *Justify all of your answers.*
- You are allowed to bring in one double-sided letter size sheet of paper and nothing else.
- 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
1	
2	
3	
4	
5	
Total	

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. [10] 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.

   (a) $T(n) = 7T\left(\frac{n}{2}\right) + n^3 \log n$

   $\log_2 7 < \log_2 8 = 3$. Let $\epsilon = \log_2 8 - \log_2 7$. Then $n^3 \log n \in \Omega\left(n^{[\log_2 7 + \epsilon]}\right)$, so we want to apply case 3 of the Master Theorem. Now $7\left(\frac{n}{2}\right)^3 \log\left(\frac{n}{2}\right) < 7\left(\frac{n}{2}\right)^3 \log n = \frac{7}{8}n^3 \log n < n^3 \log n$, so the regularity condition holds for $\delta = \frac{7}{8}$. So $T(n) \in \Theta\left(n^3 \log n\right)$.

   (b) $T(n) = 2T\left(\left\lfloor \frac{3n}{4} \right\rfloor\right) + \sqrt{n}$

   $\log_{\frac{4}{3}} 2 > \log_{\frac{4}{3}}\left(\frac{4}{3}\right) = 1$. Let $\epsilon = \log_{\frac{4}{3}} 2 - \log_{\frac{4}{3}}\left(\frac{4}{3}\right)$. Then $\sqrt{n} \in O(n) = O\left(n^{[\log_{4/3} 2] - \epsilon}\right)$ and case 1 holds. So $T(n) \in \Theta\left(n^{\log_{4/3} 2}\right)$.

   (c) $T(n) = T\left(\left\lfloor \frac{n}{2} \right\rfloor\right) + T\left(\left\lceil \frac{n}{2} \right\rceil\right) + n$

   The Master Theorem does not hold because there are two recursive calls with different terms. Gathering the terms is technically wrong (full marks).

   Bonus marks: We can bound $S(n) \leq T(n) \leq U(n)$, where $S(n) = 2S\left(\left\lfloor \frac{n}{2} \right\rfloor\right) + n$ and $U(n) = 2U\left(\left\lceil \frac{n}{2} \right\rceil\right) + n$, both of which resolve to $\Theta(n \log n)$ by the Master Theorem. The correctness of the inequalities is a straightforward inductive proof. So $T(n) \in \Theta(n \log n)$.

2. [5] You are given an array of n integers and need to find which elements of the array are equal to a given number x. A straightforward algorithm to solve this problem is to compare every element of the array to x, which takes n comparisons. Perhaps, if we were clever enough, we could compare array elements to each other to reduce the total number of comparisons. Use a decision tree argument to argue that this is a waste of time: that is, any comparison based algorithm solving this problem needs to use $\Omega(n)$ comparisons in the worst case.

Each element of the array is either equal to x or not equal to x, which gives us 2^n possible outcomes. Hence we need 2^n leaves. A decision tree of height h has at most 2^h leaves. Therefore the height h must be at least n. The height of the tree represents the worst case number of comparisons. Hence we need $\Omega(n)$ comparisons.

3. [5] Let $f(n) : \mathbb{N} \rightarrow \mathbb{R}^+$, $g(n) : \mathbb{N} \rightarrow \mathbb{R}^+$, and $h(n) : \mathbb{N} \rightarrow \mathbb{R}^+$ be such that $f(n) + g(n) \in \Theta(h(n))$. Show that $5 \max\{f(n), g(n)\} - 3 \min\{f(n), g(n)\} \in \Theta(h(n))$.

There are many acceptable solutions, including using that $\max\{f(n), g(n)\} \in \Theta(f(n) + g(n))$, which was shown in the tutorials. We argue directly from the definitions.

We note that $2 \max\{f(n), g(n)\} \geq f(n) + g(n)$. So $5 \max\{f(n), g(n)\} - 3 \min\{f(n), g(n)\} \geq 5 \max\{f(n), g(n)\} - 3 \max\{f(n), g(n)\} = 2 \max\{f(n), g(n)\} \geq f(n) + g(n)$, for all n. So $5 \max\{f(n), g(n)\} - 3 \min\{f(n), g(n)\} \in \Omega(f(n) + g(n))$.

Also note that $5 \max\{f(n), g(n)\} - 3 \min\{f(n), g(n)\} \leq 5 \max\{f(n), g(n)\} \leq 5(f(n) + g(n))$, for all n. So $5 \max\{f(n), g(n)\} - 3 \min\{f(n), g(n)\} \in O(f(n) + g(n))$.

4. [5] Show that $\sum_{i=1}^{n} i^k \in \Theta\left(n^{k+1}\right)$ directly from the definition of Θ using basic algebra.

Upper bound:

$$\sum_{i=1}^{n} i^k \quad \leq \quad \sum_{i=1}^{n} n^k = n \cdot n^k = n^{k+1}$$

Lower bound:

$$\begin{aligned}
\sum_{i=1}^{n} i^k &= \sum_{i=1}^{\lfloor n/2 \rfloor} i^k + \sum_{i=\lceil n/2 \rceil}^{n} i^k \\
&\geq \sum_{i=1}^{\lfloor n/2 \rfloor} 0 + \sum_{i=\lceil n/2 \rceil}^{n} \left\lceil \frac{n}{2} \right\rceil^k \\
&= \left(n - \left\lceil \frac{n}{2} \right\rceil + 1\right) \left\lceil \frac{n}{2} \right\rceil^k \\
&= \left(\left\lfloor \frac{n}{2} \right\rfloor + 1\right) \left\lceil \frac{n}{2} \right\rceil^k \\
&\geq \left\lceil \frac{n}{2} \right\rceil \left\lceil \frac{n}{2} \right\rceil^k \geq \left(\frac{n}{2}\right)^{k+1} = \frac{1}{2^{k+1}} n^{k+1}
\end{aligned}$$

5. [5] Explain how using unlimited (but finite) time and space you can preprocess a sorted array of n distinct integers so that subsequent queries of the form "Where is integer x in the array?" can be answered in $\Theta(1)$ time.

Do a counting sort, but instead of storing the count, which will be 1 or zero, store the array index or -1. Then we can locate a number in the array by looking in the appropriate bucket and returning the index.