

CMPUT 204 (A1)
Midterm 1: Version A
October 1, 2004

Department of Computing Science
University of Alberta

Instructor: M. Bowling

Time: 50 Minutes

Last Name:_____

First Name:_____

- Put your name above (no ID). Print clearly.
- There are 3 problems, on 3 pages (50 marks in total).
- Closed Book: Only Pen, Pencil & Eraser allowed.
- The last page is a sheet of useful facts that you may use freely.
- Use space below the questions to write your answers.

Do NOT write on this page, for instructor's use only.

total marks	question	your marks
15	1	
20	2	
15	3	
50		

Last Name: _____

ID#: _____

1. **(15 Marks)** Prove asymptotic upper and lower bounds (e.g., Θ -notation) for each of the following recurrences. Assume that in each case, the function is some positive constant for sufficiently small n . You may use the master theorem or iterated substitution, but must show how it applies. If you use iterated substitution, you do not need to prove your closed-form is correct.

1. $T(n) = 8T(\frac{n}{2}) + n^3$

2. $T(n) = 4T(\frac{n}{4}) + \sqrt{n}$

3. $T(n) = 2T(n-1) + 1$

Solution:

(a) Master theorem applies with $a = 8$, $b = 2$, and $c = 3$. Since $\log_b a = 3 = c$,

$$T(n) \in \Theta(n^3 \log n)$$

(b) Master theorem applies with $a = 4$, $b = 4$, and $c = 1/2$. Since $\log_b a = 1 \geq 1/2 = c$,

$$T(n) \in \Theta(n).$$

(c) Master theorem doesn't apply. Iterated substitution gives,

$$\begin{aligned} T(n) &= 2T(n-1) + 1 \\ &= 2(2T(n-2) + 1) + 1 \\ &= 2^2T(n-2) + 2 + 1 \\ &= 2^2(2T(n-3) + 1) + 2 + 1 \\ &= 2^3T(n-3) + 2^2 + 2 + 1 \\ &= 2^nT(0) + 2^{n-1} + \dots + 2^1 + 2^0 \\ &= 2^nT(0) + \sum_{i=0}^{n-1} 2^i \\ &= 2^nT(0) + 2^n - 1 \\ &= 2^n(T(0) + 1) - 1 \\ &\in \Theta(2^n) \end{aligned}$$

Last Name: _____

ID#: _____

2. (20 Marks) Below is pseudocode for a procedure to calculate the length of a vector stored as an array. Use a loop invariant (clearly stating the invariant) to prove the procedure is correct, i.e., that it returns $\sqrt{\sum_{i=1}^n A[i]^2}$.

```
VECTORENGTH(A)
1  s ← 0
2  for i ← 1 to length(A)
3      do s ← s + A[i]2
4  return √s
```

Solution:

Loop Invariant: $s = \sum_{j=1}^{i-1} A[j]^2$.

Initialization: Before the loop begins, $i = 1$ and $s = 0$, and the summation is empty, thus equals 0, satisfying the loop invariant.

Maintenance: By our loop invariant, at the beginning of the loop,

$$s = \sum_{j=1}^{i-1} A[j]^2.$$

At the end of the loop, $A[i]^2$ is added to s , giving,

$$\begin{aligned} s &= \sum_{j=1}^{i-1} A[j]^2 + A[i]^2 \\ &= \sum_{j=1}^i A[j]^2 \end{aligned}$$

After incrementing i we have,

$$s = \sum_{j=1}^{i-1} A[j]^2,$$

which is our loop invariant.

Termination: Let $n = \text{length}(A)$. The loop will terminate when $i = n + 1$. By the loop invariant, when it terminates, $s = \sum_{i=1}^n A[i]^2$ and $\sqrt{\sum_{i=1}^n A[i]^2}$ is returned, and thus the algorithm is correct. $\square$.

3. **(15 Marks)** Indicate, for each pair of expressions $h(n)$ and $f(n)$ in the table, if $h(n)$ is an element of O , o , Ω , ω , or Θ of $f(n)$. Your answer should be written into the table by writing “yes” or “no” in each box. Assume $c > 1$. The first row is done for you.

	$h(n)$	$f(n)$	O	o	Ω	ω	Θ
Ex.	$3n^2$	$6n \lg n$	no	no	yes	yes	no
(a)	$n(\lg n)^c$	n^2	yes	yes	no	no	no
(b)	2^{3n}	6^n	no	no	yes	yes	no
(c)	$2 + \frac{1}{n}$	1	yes	no	yes	no	yes
(d)	$(3 + c)^n$	2^{n^2}	yes	yes	no	no	no
(e)	$n^2 + (-1)^n n$	$n\sqrt{n}$	no	no	yes	yes	no

Solution: Only the “key” relationship is shown below. The rest follow immediately from this result.

(a) For all $c > 1$, $(\lg n)^c \in o(n)$. Therefore $n(\lg n)^c \in o(n^2)$.

(b) $2^{3n} = (2^3)^n = 8^n \in \omega(6^n)$.

(c) $\lim_{n \rightarrow \infty} \left(2 + \frac{1}{n}\right) = 2$. Therefore $2 + \frac{1}{n} \in \Theta(1)$.

(d) We can switch the base,

$$\begin{aligned} (3 + c)^n &= 2^{\lg(3+c)^n} \\ &= 2^{n \lg(3+c)} \end{aligned}$$

The limit of the ratio ratio of $h(n)/f(n)$ is then,

$$\lim_{n \rightarrow \infty} \frac{2^{n \lg(3+c)}}{2^{n^2}} = \lim_{n \rightarrow \infty} 2^{n \lg(3+c) - n^2}$$

Since n^2 grows faster than cn , $cn - n^2$ goes to $-\infty$ as n goes to ∞ . Thus the above limit goes to 0, so,

$$(3 + c)^n \in o(2^{n^2}).$$

(e) $n^2 + (-1)^n n \geq n^2 - n \in \Theta(n^2)$. But $n^2 \in \omega(n\sqrt{n})$, so $n^2 + (-1)^n n \in \omega(n\sqrt{n})$.

Notes on Logarithms

- Definition of $\log_b n$ ($b, n > 0$): $b^{\log_b n} = n$
- $\log_b n$ as a function in n : increasing, one-to-one
- $\log_b 1 = 0$
- $\log_b x^p = p \log_b x$
- $\log_b(xy) = \log_b x + \log_b y$
- $x^{\log_b y} = y^{\log_b x}$
- $\log_b x = (\log_b c)(\log_c x)$
- $\ln n = \log_e n$ (natural logarithm)
- $\lg n = \log_2 n$ (base 2, binary)
- $\Theta(\log_b n) = \Theta(\log_{\{\text{whatever positive}\}} n) = \Theta(\log n)$
- $\frac{d}{dx} \ln x = \frac{1}{x}$
- $(\log n)^k \in o(n^\epsilon)$, for any positives k and ϵ

Master Theorem (Simple Version)

Let $a \geq 1$ and $b > 1$ be constants, let $f(n) \in \Theta(n^c)$ for some positive c , and let $T(n)$ be defined by the recurrence

$$T(n) = aT\left(\frac{n}{b}\right) + f(n).$$

Then $T(n)$ can be bounded asymptotically as follows:

- if $c = \log_b a$, then $T(n) \in \Theta(n^c \log n)$,
- if $c > \log_b a$, then $T(n) \in \Theta(n^c)$,
- if $c < \log_b a$, then $T(n) \in \Theta(n^{\log_b a})$,

Common Summations

- $\sum_{i=1}^n i = \frac{n(n+1)}{2}$
- $\sum_{i=0}^n 2^i = 2^{n+1} - 1$
- $\sum_{i=0}^n 3^i = \frac{3^{n+1} - 1}{2}$