Name: ____________________ Net ID: ____________________ Univ. ID # ____________________

# Midterm Exam 2, CS 173, Spring 2003
Tuesday, April 8, 2003

## EXAM FORM A

**Read the following instructions before you start to work.**

- **This is a closed-book, closed-notes exam!** No calculators, paper, notebooks, or anything else. If you brought anything with you other than pencils, erasers, or pens, please leave them at the front of the room with the proctor.
- LEGIBLY print your name, Net-ID, and University ID in the slots above, and on the top of the last page.
- Fill in the administrative portion of your computer scan sheet. **MAKE SURE YOU FILL IN WHETHER YOU HAVE EXAM FORM "A" or "B".**
- There are 20 questions and 6 pages. Make sure you have a complete exam. All questions except the two on the last page should be answered on the computer scan sheet. For problem numbers 19 and 20, write your solution in the space provided, using the backs of pages if you need extra space.
- Read each question carefully and make sure you understand exactly what it is asking. If you give a beautiful answer to the wrong question, you'll get no credit. Simplify your answers as much as possible. Correct but unnecessarily complicated answers might not receive full credit.
- For reasons of consistency, the proctors will only answer syntactic questions about the exam (e.g. is that an "a" or an "e"). If you think the question is ambiguous, make and state your reasonable assumptions and answer the question.
- Don't spend too much time on any single problem. You have a whopping 150 minutes, and there are 50 points on the exam, so if a problem looks like it will take you much longer in minutes than three times its point value, you might leave it for later. Remember, *this exam is curved.*
- There is no penalty for incorrect answers.
- For a question requiring a big-Oh answer, choose the best (smallest) function that applies. For example, although $13x^2 + 2x$ is both $O(x^2)$ and $O(x^3)$, the better answer is $O(x^2)$.

For problems 1-4 (2 points each), pay particular attention to the domain and co-domain of the function described. (Either $\mathbb{N}$ (the natural numbers), or $Z$ (the integers).)
For each problem, fill in the correct circle on your bubble sheet as follows:

**(a)** if the function is both one-to-one and onto

**(b)** if the function is one-to-one but not onto

**(c)** if the function not one-to-one but is onto

**(d)** if the function is neither one-to-one nor onto

1. $f : Z \to Z, f(n) = \lfloor \frac{n}{2} \rfloor - 3$
2. $f : Z \to \mathbb{N}, f(n) = -2n$ if $n \leq 0$, and $f(n) = 2n - 1$ otherwise.
3. $f : Z \to \mathbb{N}, f(n) = -2n$ if $n \leq 0$, and $f(n) = 2n + 1$ otherwise.
4. $f : \mathbb{N} \to Z, f(n) = -\frac{n}{2}$ if $n$ is even, and $f(n) = \frac{n-1}{2}$ otherwise.

For problems 5-7 (2 points each), fill in the correct circle on your bubble sheet as follows:

**(a)** if the set described is finite and nonempty.

**(b)** if the set described is countably infinite.

**(c)** if the set described is uncountably infinite.

**(d)** if none of the above hold.

5. The set of all real numbers between $\sqrt{2}$ and $\sqrt{3}$
6. The set $\mathbb{N} \times S$, where $S$ is a finite nonempty subset of $\mathbb{N}$
7. The set of all books, where a book is defined as any finite sequence of chapters, a chapter defined as any finite sequence of paragraphs, a paragraph defined as any finite sequence of sentences, a sentence defined as any finite sequence of characters, and a character is an element of a given finite set of symbols $A$.

8. (2 points) Let $f(n) = 1 + 2 + 3 + \cdots + \frac{n}{3}$. Then $f(n) =$

(a) $O(\log n)$ (b) $O(n \log n)$ (c) $O(n^{\log_3 2})$ (d) $O(n^2)$ (e) $O(n^3)$

9. (2 points) Let $f(n) = 1 \cdot 2 \cdot 3 + 2 \cdot 3 \cdot 4 + \cdots + (n-2) \cdot (n-1) \cdot n$. Then $f(n) =$

(a) $O(n^2)$ (b) $O(n^3)$ (c) $O(n^4)$ (d) $O(n^5)$ (e) none of the above

10. (2 points) Let $f(n) = (\log 2^{\log_3 n})(3n^2)$ Then $f(n) =$

(a) $O(n^2)$ (b) $O(n^2)\log(\log n)$ (c) $O(n^2 \log n)$ (d) $O(n^3)$ (e) $O(n^3 \log n)$

11. (2 points) Suppose you have an algorithm for solving any homework assignment and that it runs in time $f(n)$ where $n$ is the number of problems. Assuming that you'd like your algorithm to run as fast as possible, which of the following would you be happiest to know?

(a) $f = O(n^2)$ (b) $n^2 = \omega(f)$ (c) $f = \Theta(n^2)$ (d) $n^2 = o(f)$

12. (3 points) Consider the following program segment:

**procedure confuse** (positive integers $m$, $n$)
    $y$ :=1
    **while** ($y < n$)
    $z$ :=$m^3$
        **while** ($z > 1$) $z := z - 2m$ **end while**
    $y := 2y$
    **end while**
**end confuse.**

The best time complexity bound is:
(a) $O(\log mn)$ (b) $O(m \log n)$ (c) $O(m^2 \log n)$ (d) $O(m^3 \log n)$ (e) $O(m2^n)$

13. (2 points) Which of the following correctly completes the inductive definition of the sequence $a_1, a_2, \ldots$, whose $n$-th term is $n^2$?

$a_1 = 1$, and for $n > 1$,

(a) $a_n = a_{n-1} + 2n - 1$
(b) $a_n = (a_{n-1})^2 + 2n - 1$
(c) $a_n = a_{n-1} + 2n + 1$
(d) $a_n = (a_{n-1})^2 + 2n + 1$
(e) $a_n = (a_{n-1} + 1)^2$

Problems 14-16 involve a method for evaluating polynomials. As an example, consider the polynomial function $f(x) = 5x^3 + 3x^2 - 6x + 7$. Then it is easily seen that $f(x) = ((5x + 3)x - 6)x + 7$. More generally, if

$$f(x) = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0,$$

then

$$f(x) = ((((a_n x + a_{n-1})x + a_{n-2})x + \cdots)x + a_1)x + a_0.$$

The following iterative algorithm uses this method to determine the value of $f(x)$, given coefficients $a_n, a_{n-1}, \ldots, a_0$ for $f$, and the value of the input $x$.

```
procedure Polly (x, a_0, a_1, a_2, ..., a_n : real numbers)
        y := ___B___
        for i := 1 to n
                y :=___C___
        end for
        output y.
end Polly.
```

14. (3 points) What two values should you substitute for B and C above so that the algorithm is correct?

    **A.** B $= a_0$, and C $= yx + a_i$

    **B.** B $= a_0$, and C $= yx + a_{n-i}$

    **C.** B $= a_n$, and C $= yx + a_i$

    **D.** B $= a_n$, and C $= yx + a_{n-i}$

    **E.** B $= a_0$, and C $= yx + a_{i+1}$

15. (2 points) To prove PolyEval is correct, we prove by induction on $k$ that for $k$ between 1 and $n$, after the $k$-th iteration of the for loop, the value of $y$ is:

    **A.** $a_n x^k + a_{n-1} x^{k-1} + \cdots + a_{n-k}$

    **B.** $a_n x^k + a_{n-1} x^{k-1} + \cdots + a_k$

    **C.** $a_n x^n + a_{n-1} x^{n-1} + \cdots + a_{n-k} x^{n-k}$

    **D.** $a_k x^k + a_{k-1} x^{k-1} + \cdots + a_1 x + a_0$

    **E.** $a_n x + a_{n-k}$

16. (3 points) Let $y_i$ denote the value of $y$ after $i$ iterations. The key property that is used in the inductive proof is that $y_{k+1} = y_k x + \mathbf{Q}$. Which is the best choice of $\mathbf{Q}$?

    (a) $a_0$ (b) $a_n$ (c) $a_n x$ (d) $a_{n-k-1}$ (e) $a_{k+1}$

Consider the following recursive algorithm $D$, which takes as input a string of characters over some finite alphabet $A$.

**procedure** D($a_1a_2\ldots a_n$: string-of-characters)
  **if** $n = 1$, **then** return $a_1$
  **else if** $n = 2k$ for some $k$, **then**
    return $\mathrm{D}(a_{k+1}a_{k+2}\ldots a_n)\mathrm{D}(a_1a_2\ldots a_k)$
  **else if** $n = 2k+1$ for some $k$, **then**
    return $\mathrm{D}(a_{k+2}a_{k+3}\ldots a_n)a_{k+1}\mathrm{D}(a_1a_2\ldots a_k)$.
**end D**

Notice that D($x$)D($y$) denotes the operation of concatenating the strings $x$ and $y$, not some form of arithmetic multiplication.

17. (2 points) What is the value returned by D(armadillo)?

    **(a)** dilloarma

    **(b)** ollidamra

    **(c)** llodimaar

    **(d)** $a_1a_2a_3a_4a_5a_6a_7a_8$

    **(e)** none of the above

18. (2 points) Let $T(n)$ be the time needed to run algorithm D on an input string of length $n$. Then $T(n)$ satisfies which of the following recurrence equations? (Below, $c$ is some constant.)

    **(a)** $T(n) = 2T(n/2) + c$

    **(b)** $T(n) = 2T(n/2)$

    **(c)** $T(n) = 4T(n/2) + c$

    **(d)** $T(n) = 2T(n-1) + c$

    **(e)** $T(n) = T(n/2) \times T(n/2)$

19. (3 points)
A string made up of parentheses is defined to be balanced if it is possible to match the left-parentheses one-to-one with the right-parentheses so that the left of the pair appears before the right of the pair, and no two matched pairs "cross" – that is, either one pair is nested within the other, or they do not overlap at all. This is hard to say, but is easy to exemplify: Examples of balanced strings of parentheses include (), (()), ()(), (()())(()). Examples of unbalanced parentheses include ())(, (, )(, )(())(.

Give an inductive definition of the set of all strings of balanced parentheses of length 0 or more.

20. 8 points total)
(a) (2 points)
Write a recursive algorithm that finds the min and max of a given set of numbers by breaking the set into two roughly equal sized pieces, and finding the overall min and max from the min and max of each half.

(b) (4 points)
Prove that your algorithm is correct by induction.

(c) (2 points)
Write a recurrence equation showing its running time. (You need not solve it).