

Practice Midterm Exam: CS103B

Do not read this exam right now!

The best way to use this exam for preparation is to do the following first:

- 1) Organize your notes and handouts. This exam covers algorithm analysis, sets, relations and trees.
- 2) Read through them all, indexing them so you can go directly to definitions and examples as needed.
- 3) Review your problem sets and do additional problems (see below) to help in areas where you are having trouble.
- 4) The exam will be time-pressured so it's important that you do not have to look everything up. There is no need to memorize anything, but you want to be able to work fast.

Once you feel you have the material under control, set aside 1.5 hours and find a quiet place to take this exam.

Here are the rules:

- This is an **open-note** exam only. Allowed: class handouts, problems sets, problem set solutions and your own notes. Not allowed: any textbook.
- The exam consists of two sections: short answer and problems. Try to split your time evenly between the two.
- Always show your work so we can give you partial credit.

Good luck!

Additional Practice Problems

- 1) Give big-oh estimates for the factorial function $f(n) = n!$, and the log of the factorial function.
- 2) Give a big-oh estimate for $f(n) = 3n \log(n!) + (n^2+3) \log n$.
- 3) Show that the following algorithm determines the number of 1 bits in the bit string S .

```
bitcount(S: bit string) {
    count = 0;
    while (S ≠ 0) {
        count = count + 1;
        S = S ^ (S-1);
    }
} // count contains the number of 1's in S
```

$S-1$ is the bit string obtained by changing the rightmost one bit of S to 0 and all the zero bits to the *right* of this to 1's.

How many AND operations are needed to find the number of 1 bits in S ?

- 4) Symmetric difference of sets A and B denoted, $A \oplus B$ is the set containing those elements in either A or B , but not both in A and B .

Prove or disprove: $(A \oplus B) \oplus B = A$. If you prove this, do so using a containment proof.

- 5) Suppose that A , B and C are sets such that $A \oplus C = B \oplus C$. Must it be the case that $A = B$?
- 6) A relation is called circular if $a R b$ and $b R c$ imply $c R a$. Prove or disprove: R is reflexive and circular iff it is an equivalence relation.
- 7) A B -tree of degree k is a rooted tree such that all its leaves are at the same level, its root has at least two and at most k children (unless it is a leaf), and every internal vertex other than the root has $\lceil k/2 \rceil$ but no more than k children.

Give an upper bound and lower bound for the number of leaves in a B -tree of degree k and height h .

Solutions to Practice Problems

- 1) Big-oh for $n!$ can be obtained by noting that each term in the product $1 * 2 * \dots * n$ does not exceed n . So,

$$\begin{aligned}n! &= 1 * 2 * \dots * n \\ &\leq n * n * \dots * n \\ &= n^n\end{aligned}$$

So $n!$ is $O(n^n)$. Taking log of both sides of the inequality established for $n!$ we obtain:

$$\log n! \leq \log(n^n) = n \log n. \text{ Therefore, } \log n! = O(n \log n)$$

- 2) The first term is $O(n^2 \log n)$ (see #1). The second term is $O(n^2 \log n)$. By the sum rule, big-oh for the expression is $O(n^2 \log n)$.
- 3) By the way that $S-1$ has been defined, we can see that $S \wedge (S-1)$ is that same as S except that the rightmost one bit has been changed to 0. Thus, we add 1 to count for every one bit (since we stop when $S = 0$, i.e., as soon as S consists only of 0 bits). The number of AND operations is equal to the value of count. We perform the AND for each of the one bits in S .
- 4) First we have to prove that $A \subseteq (A \oplus B) \oplus B$. Suppose $x \in A$. If $x \in B$, then $x \notin A \oplus B$ by the definition of symmetric difference. If we then take $(A \oplus B) \oplus B$ (when $x \in A$ and $x \in B$) then $x \in (A \oplus B) \oplus B$ since this would consist of the elements of A not in B , along with the elements both sets have in common. If $x \notin B$, then $x \in A \oplus B$ so again $x \in (A \oplus B) \oplus B$. Now we have to prove that $(A \oplus B) \oplus B \subseteq A$. Suppose $x \in (A \oplus B) \oplus B$. There are two cases: If $x \notin B$, then $x \in A \oplus B$, and therefore $x \in A$. If $x \in B$, then $x \notin A \oplus B$, and the only way for that to happen is for x to be in A .
- 5) Yes. Proof by contradiction: Suppose $x \in A$, but $x \notin B$. If $x \in C$, then $x \notin (A \oplus C)$ but $x \in (B \oplus C)$ which is a contradiction. If $x \notin C$, then $x \in (A \oplus C)$ but $x \notin (B \oplus C)$ which is also a contradiction. So we have shown that $A \subseteq B$ and $B \subseteq A$ and therefore, $A = B$.
- 6) Suppose R is reflexive and circular. We need to show that R is symmetric and transitive. Let $(a,b) \in R$. Since $(b,b) \in R$ it follows by circularity that $(b,a) \in R$. This proves symmetry. Now if $(a,b) \in R$ and $(b,c) \in R$ then by circularity $(c,a) \in R$ and then by symmetry, $(a,c) \in R$. Thus R is transitive. Conversely, transitivity and symmetry immediately imply circularity and reflexivity (e.g., symmetry provides (b,a) for each (a,b) and transitivity provides (b,b)). Thus, every equivalence relation is reflexive and circular.

7) Leaves upper bound: k^h ; lower bound: $2 \cdot \text{ceiling}(k/2)^{h-1}$

Practice midterm

Short Answer Section: 50 points total (point value in parentheses)

1) Give a tight big-oh bound on the following three functions:

a) (2) $(3n - 8 - 4n^3) / (2n - 1)$

b) (4) $1^3 + 2^3 + \dots + n^3$

c) (2) $\text{ceil}(n + 2) * \text{ceil}(n/3)$

2) (2 each) Is the relation R on $\{1, 2, 3, \dots\}$ where aRb means $a \mid b$

Reflexive?	_____
Symmetric?	_____
Anti-symmetric?	_____
Transitive?	_____

3) (4) Prove or disprove for sets A, B, C:

$$(A-C) - (B-C) = A - B$$

4) (4) A tree has 999 vertices; how many edges does it have?

5) (4) How many non-isomorphic trees are there with four vertices? (Isomorphic trees have exactly the same form, i.e., there is a one-to-one correspondence between their vertex sets that preserves edges.) Draw them.

6) (4) How many full 3-ary trees are there with 6 vertices?

7) (4) Heap sort has what complexity?

8) (2) Suppose $|A| = n$. Find the number of binary relations on A.

9) True or False (4 points each)

a) There is a set A such that $|P(A)| = 12$. _____

b) If two trees have the same number of vertices and the same degrees, then the trees are isomorphic. _____

c) If $A \cap B = A \cup B$ then $A = B$. _____

Problem Section

1) (20 points) Prove or disprove: The transitive closure of the symmetric closure of a relation is the same as the symmetric closure of the transitive closure of this relation.

2) (30 points) The rooted Fibonacci trees T_n are defined recursively as follows: T_1 and T_2 are both the rooted tree consisting of a single vertex; for $n > 2$, the rooted tree T_n is constructed from a root with T_{n-1} as the left subtree and T_{n-2} as the right subtree.

a) Draw T_1 , T_2 , T_3 , T_4 , and T_5 . (10 points)

b) Define a recurrence relation for the number of vertices in a rooted Fibonacci tree T_n . (5 points). Put your answer in the box.

c) Define a **closed form** formula for this recurrence relation. Prove your answer is correct using induction. (You may use F_n (the Fibonacci sequence) in your formula.) (15 points)

Practice Midterm Solutions

1a) $(3n - 8 - 4n^3) / (2n - 1) \leq (3n^3 + 8n^3 + 4n^3) / (2n - n) = 15n^3/n = 15n^2$; $O(n^2)$

1b) $1^3 + 2^3 + \dots + n^3 \leq n^3 + n^3 + \dots + n^3 = n * n^3 = n^4$; $O(n^4)$

1c) $O(n^2)$

2) yes, no, yes, yes

3) This is false

4) 998

5) there are 4 such trees.

6) 0

7) $O(n \log n)$

8) $A \times A$ has n^2 elements, so the power set of $A \times A$ has size $2^{(n^2)}$

9a) F

9b) F

9c) T

1) counterexample: $R = \{(a,b), (a,c)\}$. Transitive closure of the symmetric closure is $\{(a,a), (a,b), (a,c), (b,a), (b,b), (b,c), (c,a), (c,b), (c,c)\}$. This is not the same as the symmetric closure of the transitive closure: $\{(a,b), (a,c), (b,a), (c,a)\}$.

2b) $v_1 = 1, v_2 = 1; v_n = v_{n-1} + v_{n-2} + 1$ (the "+1" is for the root) for $n \geq 3$

2c) $2f_n - 1$ where f_n is the n th Fibonacci number.

$P(n)$ denotes: a closed form formula for $v_1 = 1, v_2 = 1; v_n = v_{n-1} + v_{n-2} + 1$ is $2f_n - 1$

where f_n is the n^{th} Fibonacci number.

base case: $v_1 = (2 * 1) - 1 = 1; v_2 = (2 * 1) - 1 = 1$ ($F(1) = 1, F(2) = 1$)

inductive hypothesis: Assume that for all positive numbers i where $i \leq k$: $v_i = 2f_i - 1$

and show $P(k+1)$: $v_{k+1} = 2f_{k+1} - 1$

PROOF:

$$v_{k+1} = v_k + v_{k-1} + 1$$

from the recurrence relation

$$v_{k+1} = (2f_k - 1) + (2f_{k-1} - 1) + 1$$

substitute inductive hypothesis

$$v_{k+1} = 2f_k + 2f_{k-1} - 1 + 1 - 1$$

$$v_{k+1} = 2f_k + 2f_{k-1} - 1$$

$$v_{k+1} = 2f_{k+1} - 1$$

from the recurrence relation

$P(k+1)$ is true when $P(k)$ is true, and therefore $P(n)$ is true for all integers $n \geq 1$.

