MATH 160 **Sample Final Exam Questions**

You may use your books and notes for the exam. Give *complete*, *concise*, *clear*, and *orderly* solutions to all of the problems, and *justify all of your answers*. The actual exam will be shorter than this.

1. There might be True/False questions, with partial credit for brief justifications. (Sample questions omitted).
2. Give a truth table for $a \vee (b \to a)$. Repeat for $(a \to (b \vee c)) \leftrightarrow b$
3. Use the procedure from the textbook to convert $(p \to q) \wedge \neg r$ into disjunctive normal form. Repeat for $(a \to (b \vee c)) \leftrightarrow b$
4. Is $(p \wedge q) \to p$ a tautology? If it is not a tautology, give a very clear argument as to why it is not. If it is a tautology, prove it in two different ways. Repeat for $(p \to q) \to (\neg p \to \neg q)$. Repeat for $(p \to q) \to (q \to p)$.
5. You are caught between your well-meaning guidance counselor (who writes your recommendation letters) and your financial aid officer (who controls your scholarship). They are debating how to determine which students should receive a new scholarship. They both agree that the scholarship should go to students who: are seniors, have a GPA of at least 3.5, and participate in at least one extracurricular activity. They have access to a full list of students and their information.
   (a) Your guidance counselor says:"Let's make three separate lists: all seniors, all students with a GPA of at least 3.5, and all students who are in at least one extracurricular activity. Then we combine those three lists (removing duplicates), and award the scholarship to everyone on the combined list." Is your guidance counselor correct? If not, explain why. Describe, in simple terms, who actually ends up receiving the scholarship using set logic or examples.
   (b) Your financial aid officer says: "Let's first make three separate lists: all students who are not seniors, all students with a GPA below 3.5, and all students who are in no extracurricular activities. Then we find the students who appear on all three of these lists. Finally, we remove those students from the full list and award the scholarship to whoever remains." Is the financial aid officer correct? If not, explain why. Describe who ends up receiving the scholarship.
   (c) What is a more simple (and correct) way of creating the desired list, if there is one?
6. Compute $\lfloor 3.1345 \rfloor$, $\lceil 1.234 \rceil$, and 34 mod 11.
7. If $f(x) = x+3$ is defined on the real numbers, is it one-to-one? Onto? Invertible? Justify your answers. Repeat for $f(x) = x^2 + 7$ defined on integers.
8. Simplify $\sum_{i=2}^{n} (2^i - i^2)$
9. Use induction to prove that for all $n \geq 1$, $\prod_{k=1}^{n} \left(1 + \frac{1}{k}\right) = n + 1$. (This one is really not that hard–you just need to remember the notation!)
10. Consider the following algorithm:

```
mystery(int n):
    if(n==0):
        return 0;
    else:
        return mystery(n-1)+2*n-1;
```

    (a) What function of n does `mystery` compute? Justify.
    (b) What is the computational complexity of `mystery(n)`? Justify.
    (c) Rewrite `mystery(int n)` so that it is as efficient as possible.
11. The *range* of an array is the difference between the maximum element of the array and the minimum element of an array. Write pseudocode for an algorithm that computes the range of an array. What is the computational complexity of your algorithm? Is it possible to write an algorithm that has a lower computational complexity? Justify.

12. Give the computational complexity of the following algorithm. Make sure you fully justify your answer!

```
public static void printPairs(int n) {
    for (int i = 0; i < n; i++) {
        for (int j = i + 1; j < n; j++) {
            System.out.println(i + " " + j);
        }
    }
}
```

13. Give and solve a recurrence relation for the computational complexity of the following algorithm.

```
public static int countCalls(int n) {
    if (n <= 1) {
        return 1;
    }
    return countCalls(n - 1) + countCalls(n - 1);
}
```

14. Let `A=00001111` and `B=01010101`. Compute `~A`, `A&B`, `A|B`, and `A^B`.

15. Compute $\binom{8}{6}$ and 6! using the fewest number of operations.

16. How many passwords are possible if they consist of lower case letters (26 possible) and/or digits (10 possible) with the following constraints

    (a) 5 characters long, no other constraints.

    (b) Must be 2 letters followed by 3 digits, where letters can repeat but digits cannot.

    (c) Contains either 4 letters, none of which can be repeated, or 4 characters that can be either number or letters.

17. How many distinct ways can the letters of BALLOON be rearranged?

18. In a group of 40 students, 25 take Mathematics, 18 take Computer Science, 10 take both. How many students take at least one of the two subjects?

19. In a school, 120 students take French, 100 take Spanish, 30 take both, 40 take neither. How many students are there in total?

20. In a group of 15 people at a party, some pairs shake hands.

    (a) Prove that the number of people who shook hands with an odd number of people is even.

    (b) Prove that at least two people shook hands with the same number of people.

21. Does every vertex in Q_n have the same degree? If so, what is it? Repeat for C_n, P_n, K_n, and $K_{m,n}$.

22. Reading Question 9.25.

23. Problem 9.21 and 9.22.

24. What is the coefficient of x^{11} in $(3x - 9)^{19}$?

25. Prove that $\displaystyle\sum_{k=0}^{n} 3^k \binom{n}{k} = 4^n$.

26. Prove that the sum of the first n odd integers is n^2 by

    (a) Evaluating a summation.

    (b) Using mathematical induction.

27. Problem 6.13, 6.16. For both of these, an important detail to keep in mind is that `A.get(k)` takes $\Theta(1)$ time on an array-based list but $\Theta(k)$ time on a linked list.

28. Use induction to prove that the following program correctly computes $\sum_{k=1}^{n} k$ for all $n \geq 1$.

    You are NOT proving what the summation is. You are proving that the code correctly produces it.

```
int sum(int n) {
    if(n==1) {
        return 1;
    } else {
        return n + sum(n-1);
    }
}
```

    Also, give the complexity of this algorithm. Then come up with a more efficient algorithm to solve the problem, giving the complexity of your algorithm.

29. The following method returns true if and only if none of the entries of the array are 0. In other words, it determines the truth value of the expression $\forall x(a[x] \neq 0)$.

```
boolean noZeroElements(int[] a, int n) {
    for(int i=0;i<n;i++) {
        if(a[i] == 0 )
            return false;
    }
    return true;
}
```

    The two methods below implement a similar idea for two arrays. Assume `a` and `b` have the same size.

```
boolean unknown1(int[] a, int[] b, int n) {
    for(int i=0;i<n;i++) {
        if( a[i]==0 && b[i]==0 )
            return false;
    }
    return true;
}
```

```
boolean unknown2(int[] a, int[] b, int n) {
    if(noZeroElements(a, n)) {
        return true;
    } else if(noZeroElements(b, n) {
        return true;
    } else {
        return false;
    }
}
```

    (a) Use `a`, `b`, quantifier(s) and/or logical operators to specify what `unknown1` determines.

    (b) Use `a`, `b`, quantifier(s) and/or logical operators to specify what `unknown2` determines.

    (c) Prove or disprove that `unknown1` and `unknown2` are determining the same thing.

30. Give a tight bound on the worst-case complexity the following algorithm.

```
void foo(int n,int m) {
    int foo = 0;
    for(int i = 0 ; i < n ; i++) {
        for(int j=0;j<n/2;j++) {
            foo++;
        }
    }
    for(int j = 0 ; j < m ; j++)
        foo++;
}
```