Claire Minahan

# Final Exam

1. For each of the following, circle one answer.
   1.1. Which of the following statements is logically equivalent to $p \rightarrow q$?
   (a) $q \rightarrow p$
   (b) $\neg p \lor q$
   (c) $\neg p \rightarrow \neg q$
   (d) $p \land \neg q$

   1.2. What must be shown in order to prove a statement of the form $\forall x P(x)$?
   (a) That $P(x)$ is true for every $x$ in the domain
   (b) That $P(x)$ is true for at least one $x$ in the domain
   (c) That $P(x)$ is true for exactly one $x$ in the domain
   (d) That if $P(x)$ is true for $x = k$, then $P(x)$ is true for $x = k + 1$

   1.3. Which of the following statements is logically equivalent to $\neg\exists x(P(x) \land Q(x))$?
   (a) $\exists x(\neg P(x) \lor \neg Q(x))$
   (b) $\exists x(\neg P(x) \land \neg Q(x))$
   (c) $\forall x(\neg P(x) \land \neg Q(x))$
   (d) $\forall x(\neg P(x) \lor \neg Q(x))$

   1.4. Let $P(x)$ be the statement, “x is taking a math class”, and let $Q(x)$ be the statement, “x is a computer science major”, where the domain is all students. Which of the following represents the statement, “Every computer science major is taking a math course” ?
   (a) $\forall x(P(x))$
   (b) $\forall x(P(x) \rightarrow Q(x))$
   (c) $\forall x(Q(x) \rightarrow P(x))$
   (d) $\forall x(Q(x) \land P(x))$

   1.5. Suppose $A \cup B = A$. Which of the following must be true?
   (a) $B = \emptyset$
   (b) $A - B = A$
   (c) $B \subseteq A$
   (d) $A \cap B = \emptyset$

   1.6. What is the cardinality of the set $\{2, 4, \{8, 16\}, \emptyset\}$?
   (a) 3
   (b) 4
   (c) 5
   (d) 6

   1.7. Which of the following sets is uncountably infinite?
   (a) The set of rational numbers, Q
   (b) The set of integers, Z
   (c) The set of natural numbers, N
   (d) The set of complex numbers, C

1.8. Which of the following describes the function $f : Z \rightarrow Z$ given by $f(x) = x^2 + 1$?
(a) An injection
(b) A surjection
(c) A bijection
(d) A relation

1.9. Let $f : A \rightarrow B$ and $g : B \rightarrow A$ be functions. What are the domain and codomain of $f \circ g$?
(a) $f \circ g : A \rightarrow A$
(b) $f \circ g : A \rightarrow B$
(c) $f \circ g : B \rightarrow A$
(d) $f \circ g : B \rightarrow B$

1.10. Which step fundamentally differs in a proof by strong induction compared to one by weak induction?
(a) The Basis Step
(b) The Inductive Hypothesis
(c) The Inductive Step
(d) The Conclusion

1.11. What is not a requirement of a good algorithm?
(a) It checks for invalid inputs
(b) It produces the correct output for a given input.
(c) Its expected inputs and outputs are defined
(d) It is applicable to all inputs of the specified form.

1.12. How many bit strings of length 16 begin with 101 and end with 1101?
(a) $2^{13} + 2^{12}$
(b) $2^9$
(c) $13! + 12! – 9!$
(d) $2^{13} + 2^{12} – 2^9$

1.13. How many bit strings of length 16 begin with 101 or end with 1101?
(a) $2^{13} + 2^{12}$
(b) $2^9$
(c) $13! + 12! – 9!$
(d) $2^{13} + 2^{12} – 2^9$

1.14. Let G = (V, E) be a graph with n vertices and m edges. What is $\Sigma_{v \in V} \deg(v)$ equal to?
(a) $nm$
(b) $2m$
(c) $n + m$
(d) $n(n + 1)$

1.15. How many edges are in $K_7$, the complete graph on 7 vertices?

(a) 42
(b) 7
(c) 21
(d) 14

2. Consider the following proposition:

$(\neg p \vee q \vee s) \wedge (q \vee s \vee \neg r) \wedge (\neg p \vee q \vee \neg r) \wedge (\neg p \vee \neg q \vee r) \wedge (p \vee \neg s \vee r) \wedge (\neg s \vee \neg p \vee q)$

(a) How many rows would be in the truth table for this proposition?

**$2^n \rightarrow 2^4 = 16$**

(b) Give an assignment of truth values to propositional variables that satisfies this proposition.

**p = T, q = T, s = T, r = T**

**(F ∨ T ∨ T) ∧ (T ∨ T ∨ F) ∧ (F ∨ T ∨ F) ∧ (F ∨ F ∨ T) ∧ (T ∨ F ∨ T) ∧ (F ∨ F ∨ T)**

**= T ∧ T ∧ T ∧ T ∧ T ∧ T**

**= T**

(c) Give an assignment of truth values to propositional variables that does not satisfy this proposition.

**p = T, q = F, s = T, r = T**

**(F ∨ F ∨ T) ∧ (F ∨ T ∨ T) ∧ (F ∨ F ∨ T) ∧ (F ∨ T ∨ F) ∧ (T ∨ F ∨ F) ∧ (F ∨ F ∨ F)**

**= T ∧ T ∧ T∧ T ∧ T ∧ F**

**= F**

3. Let $f : A \rightarrow B$, $g : B \rightarrow C$, and $h : C \rightarrow D$ be functions. Prove or disprove:

If $h \circ g \circ f : A \rightarrow D$ is a surjection, then $g : B \rightarrow C$ is also a surjection.

**Let y ∈ D. Since *h* ◦ *g* ◦ *f* is a surjection, then there must exist a w ∈ A such that *h*(g(f(y))) = w. There then must exist a x ∈ B such that h(g(y)) = y, and there must exist some z such that h(y) = z. Then we have found a z such that g(x) = z.**

**Therefore, *g* : B → C is also a surjection.**

4. Consider the following statement:

Let m and n be integers. If m + n is odd, then either m or n is odd, but not both.

(a) Which proof technique should be chosen to prove this statement? Explain your answer.

**The proof technique that should be chosen is proof by contradiction because it can work with most proofs and it works with “if-then” cases.**

(b) What is a proof technique that should not be chosen to prove this statement? Explain your answer.

**A proof technique that should not be chosen to prove this statement is prove by exhaustion because it would require an infinite amount of possibilities to plug in in order to prove the statement.**

(c) Prove that, where m and n are integers, if m + n is odd, then either m or n is odd, but not both.

**Proof:**

**Assume m and n are both even. By definition, m = 2k and n = 2l, where k and l are some integers. This then means that m + n = 2k + 2l = 2(k + l) = 2x with x being some integer. By definition, this makes m + n even.**

**Therefore, by contraposition, if m + n is odd, then either m or n is odd, but not both.**

Now, consider the following statement:

Let A and B be sets. If $A \subseteq B$, then $A \cap (\overline{B} \cup \overline{A}) = \emptyset$

(d) Which proof technique should be chosen to prove this statement? Explain your answer.

**Direct proof because this is a statement in the form of "if-then".**

(e) What is a proof technique that should not be chosen to prove this statement? Explain your answer.

**A proof technique that should not be chosen to prove this statement is prove by exhaustion because it would require an infinite number of possibilities to plug in in order to prove the statement.**

(f) Prove that, where A and B are sets, if $A \subseteq B$, then $A \cap (\overline{B} \cup \overline{A}) = \emptyset$

**Proof:**

**Because $A \subseteq B$, $\forall x\ (x \in A \rightarrow x \in B)$. This means that $(B \cup A) = B$ because $(B \cup A) = \{x \in U \mid x \in A \lor x \in B\}$, but all elements of A are in B. However, the complement of B is now everything not in A and not in B, which is $\{x \in U \mid \neg(x \in B)\}$. The intersection of these two sets, A and the complement of (B ∪ A), is the same as the intersection of A and the complement of B. This is then equal to $\{x \in U \mid x \in A \land x \in \overline{B})\}$. Because we have already proven that all the element of A are in B, that means that no elements of A are in $\overline{B}$, therefore these two sets have nothing in common, leading to an empty set.**

**Therefore, if $A \subseteq B$, then $A \cap (\overline{B} \cup \overline{A}) = \emptyset$**

5. Using graph terminology, a full binary tree can be defined recursively as follows:
   *Basis Step*: A single vertex is a full binary tree, in which case that vertex is also the root.
   *Recursive Step*: If T1 and T2 are disjoint full binary trees, then there is a full binary tree consisting of a root vertex, r, with exactly the roots of T1 and T2 as its neighbors.

Let T be a full binary tree. Show that if T has n vertices and m edges, then m = n – 1

6. Using *either* the Tree Method or the Master Theorem, solve:
   (a) T (n) = 4T ( n/4) + O(n)

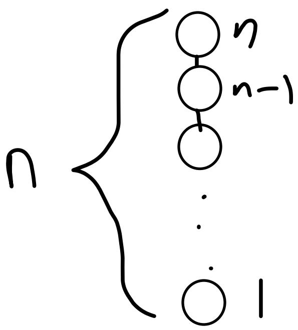
**Master Theorem:**

**a = 4, b = 4, d = 1**

**$\log_b a = \log_4 4 = 1 = d$**

**→ $O(n^4 \log n)$**

   (b) T (n) = T (n − 1) + O(n)




**$T(n) = \Sigma^{n}_{i=1} 2^i = 2^{n+1} - 1 = 2^n - 1 = O(2^n)$**

7. Give pseudocode for a recursive algorithm to find the only pair of differing elements in sequences. You may assume that they have equal lengths, but differ by exactly one element. For example, given sequences (9, −64, 21, 0) and (9, −64, 16, 0), your algorithm should return (21, 16).

---

$\textsc{DiffElements}(A \leftarrow (a_1, a_2, \ldots, a_n), B \leftarrow (b_1, b_2, \ldots, b_n))$

---

**Input: a non-empty, finite sequence of integers**
**Output: a pair of different elements in sequences**

**if n = 1, then:**
  **return ($a_1$, $b_2$)**
**else if $a_n \neq b_n$, then:**
  **return ($a_n$, $b_n$)**
**else:**
  **return DIFFELEMENTS(A ← ($a_1$, $a_2$, …, $a_{n-1}$), B ← ($b_1$, $b_2$, …, $b_{n-1}$))**

8. State and prove a lemme in order to show the correctness of your algorithm.

***Lemma*: Let A and B be non-empty, finite sequences of equal length that differ by exactly one element. If DIFFELEMENTS(A, B) returns (x, y), then x ≠ y.**

**Proof: Let and A and B be non-empty, finite sequences of equal length that differ by exactly one element. Let |A| = |B| and be denoted by n.**

***Basis Step*:**

**Let $n = 1$. Then $A = a_1$ and $B = b_1$. Because they only have one element they must differ by exactly one element, then certainly $a_1 \neq b_1$.**

***Inductive Hypothesis*:**

**Suppose, for all integers $1 \leq n \leq k$, if A and B are finite sequences of equal length with exactly one differing element and DIFFELEMENTS(A, B) returns (x, y), then $x \neq y$.**

***Inductive Step*:**

**Let $n = k + 1$. Since $k \geq 1$, $n \geq 2$. Let $a_1 = b_1$ and $(a_2, a_3, \ldots, a_n)$ be denoted A', and $(b_2, b_3, \ldots, b_n)$ be denoted B'. Then DIFFELEMENTS recurses on A' and B', which is of size $\leq k$.**

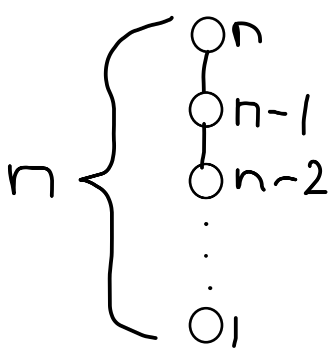
**By hypothesis, DIFELEMENTS obtains some value such that $a_i \neq b_i$.**

**If $a_i \neq b_i$, then the two sequences differ at that element. Thus DIFELEMENTS correctly returns (x, y).**

**Therefore, by PMI, DIFELEMENTS is correct.**

9. Set up and solve a recurrence relation estimating the complexity of your algorithm.

**$T(n) = T(n-1) + O(1)$**




**$\rightarrow \Sigma^{n}_{i=1} 1 = n \rightarrow$ O(n)**

10. Consider the function $f : R \rightarrow R$ given by:

$$f(n) = (n^5 + n^4 \log n)\ (2 \cdot n! + n! + 28^n)\ (\log n + 1)\ (17 \log n + 19)\ (n^{16384} + 2^n + 3)$$

   (a) Give any Big-O estimate for this function.

**$(n^5 + n^4 \log n)$: $n^5 \rightarrow O(x^5)$, $n^4 \rightarrow O(x^4)$, $\log n \rightarrow O(\log n)$ … $O(n^5)$**

**$(2 \cdot n! + n! + 28^n)$: $2n! \rightarrow O(n!)$, $n! \rightarrow O(n!)$, $28^n \rightarrow O(28^n)$ … $O(n!)$**

**$(\log n + 1)$: $\log n \rightarrow O(\log n)$, $1 \rightarrow O(1)$ … $O(\log n)$**

**$(17 \log n + 19)$: $17 \log n \rightarrow O(\log n)$, $19 \rightarrow O(19)$ … $O(\log n)$**

**$(n^{16384} + 2^n + 3)$: $n^{16384} \rightarrow O(n^{16384})$, $2^n \rightarrow O(2^n)$ … $2^n$**

**$\rightarrow O(n^5 \cdot n! \cdot \log n \cdot \log n \cdot 2^n) = O(n^6 \cdot n! \cdot (\log n)^2 \cdot 2^n)$**

(b) Give any Big-Ω estimate for this function.

**Ω(x)**

(c) Give any Big-Θ estimate for this function.

**$\Theta\ (n^5 \cdot n!\ \cdot (\log n)^2 \cdot 2^n)$**

11. Consider the 23 letters of the classical Latin alphabet:

A, B, C, D, E, F, G, H, I, K, L, M, N, O, P, Q, R, S, T, V, X, Y, Z

(a) How many strings of length 12 that include the substring "SPQR" can be formed from the classical Latin alphabet without any repetitions?

**_ _ _ _ _ _ _ _ _ _ _ _**
**SPQR …..** **23 – 4 = 19. "SPQR" is one slot, so there are now 12 – 4.**
**9 nCr 1= 9 (The 9 slots to put "SPQR" in)**
**9 * 19 * 18 * 17 * 16 * 15 * 14 * 13 * 12**
**= $2.74 * 10^{10}$**

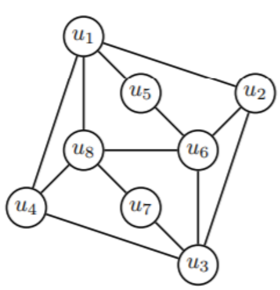
(b) How many characters are guaranteed to be the same letter within a string of length 52 that has been formed from the classical Latin alphabet?

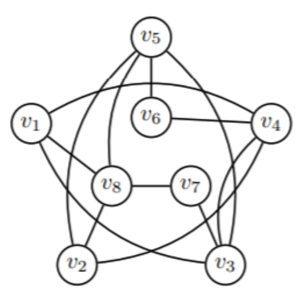
**n = 52, k = 23, → ⌈52 / 23⌉ = 2.26 → 3**

(c) Numerous modern alphabets descend from this one. How many ways can 7 letters be selected from the classical Latin alphabet for inclusion in a new alphabet?

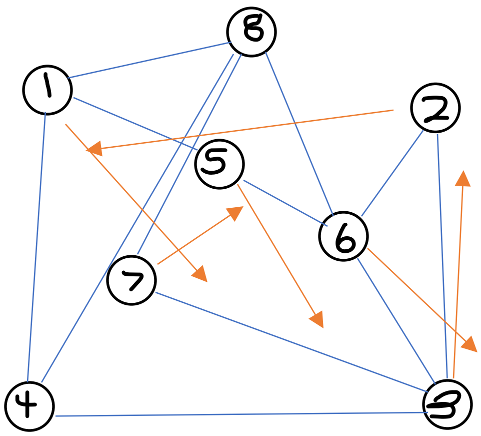
**_ _ _ _ _ _ _ → 23 * 22 * 21 * 20 * 19 * 18 * 17 = 1235591280**

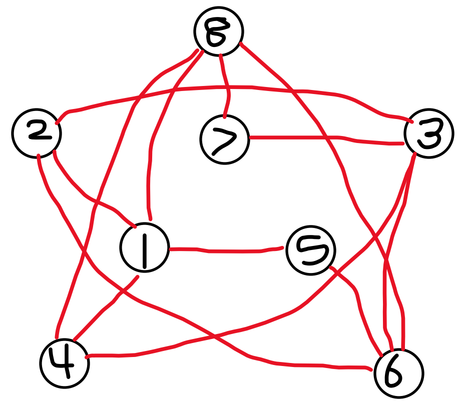
12. Consider the following simple, undirected graphs:



(a) Show that these graphs are isomorphic.




(b) Does either graph have an Eulerian path? If so, construct one; if not, explain why.

**No, there is no way to traverse every edge exactly once.**

(c) Does either graph have a Hamiltonian path? If so, construct one; if not, explain why.