Massachusetts Institute of Technology
6.042J/18.062J, Fall ’17: Mathematics for Computer Science
Prof. Albert R Meyer, Dr. Zachary Abel

October 31
revised Tuesday 24th October, 2017, 17:54

Midterm Exam October 31

Your name:____

	9:30AM	**Table (A–H):**	_______
Identify your Team:	**1PM**	**Table (A–K):**	_______
	2:30PM	**Table (A–E, 1–13):**	_______

- This exam is **closed book** except for a 2-sided cribsheet. Total time is 90 minutes.
- Write your solutions in the space provided. If you need more space, write on the back of the sheet containing the problem.
- In answering the following questions, you may use without proof any of the results from class or text.

DO NOT WRITE BELOW THIS LINE

Problem	Points	Grade	Grader
1	20		
2	20		
3	20		
4	20		
5	20		
6	20		
7	20		
8	20		
9	20		
10	20		
Total	200		

Problem 1 (**Recursive Data**) (**20 points**).
One way to determine if a string has matching brackets, that is, if it is in the set, RecMatch[1], is to start with 0 and read the string from left to right, adding 1 to the count for each left bracket and subtracting 1 from the count for each right bracket. For example, here are the counts for two sample strings:

$$\begin{array}{ccccccccccccc} & [&] &] & [& [& [& [& [&] &] &] &] \\ 0 & 1 & 0 & -1 & 0 & 1 & 2 & 3 & 4 & 3 & 2 & 1 & 0 \end{array}$$

$$\begin{array}{ccccccccccc} & [& [& [&] &] & [&] &] & [&] \\ 0 & 1 & 2 & 3 & 2 & 1 & 2 & 1 & 0 & 1 & 0 \end{array}$$

A string has a *good count* if its running count never goes negative and ends with 0. So the second string above has a good count, but the first one does not because its count went negative at the third step. The empty string λ has a good count of 0. Let

$$\text{GoodCount} ::= \{s \in \{\,]\,,[\,\}^* \mid s \text{ has a good count}\}.$$

One way to prove that RecMatch = GoodCount is to show that each includes the other.

(a) One of the inclusions

$$\text{GoodCount} \subseteq \text{RecMatch},$$
$$\text{RecMatch} \subseteq \text{GoodCount},$$

is easy to prove by structural induction, while the other inclusion follows easily by strong induction. Which inclusion is easy to prove by structural induction?

(b) State an induction hypothesis that allows an easy proof by structural induction of the inclusion from part (a). No proof is required.

(c) The other inclusion can be proved by strong induction. State a strong induction hypothesis that leads to a straightforward proof of this other inclusion. No proof is required.

[1] RecMatch is defined recursively: **Base case:** $\lambda \in$ RecMatch. **Constructor case:** If $s, t \in$ RecMatch, then $[\, s\,]\, t \in$ RecMatch.

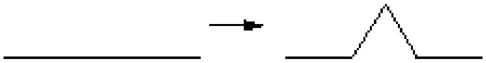


Figure 1 Constructing the Koch Snowflake.

Problem 2 (Recursive Data) (20 points).
Fractals are an example of mathematical objects that can be defined recursively. In this problem, we consider the Koch snowflake. Any Koch snowflake can be constructed by the following recursive definition.

- **Base case**: An equilateral triangle with a positive integer side length is a Koch snowflake.
- **Constructor case**: Let K be a Koch snowflake, and let l be a line segment on the snowflake. Remove the middle third of l, and replace it with two line segments of the same length $|l|$, as is done in Figure 1

  The resulting figure is also a Koch snowflake.

Prove by structural induction that the area inside any Koch snowflake is of the form $q\sqrt{3}$, where q is a rational number.

Problem 3 (Infinite Cardinality) (20 points).
Let $\{1, 2, 3\}^{\omega}$ be the set of infinite sequences containing only the numbers 1, 2, and 3. For example, some sequences of this kind are:

$$(1, 1, 1, 1...),$$
$$(2, 2, 2, 2...),$$
$$(3, 2, 1, 3...).$$

Prove that $\{1, 2, 3\}^{\omega}$ is uncountable.

Hint: One approach is to define a surjective function from $\{1, 2, 3\}^{\omega}$ to the power set $\text{pow}(\mathbb{N})$.

Problem 4 (**Diagonal Argument / Set Theory**) (**20 points**).

(a) Explain why the union of two countable sets is countable.

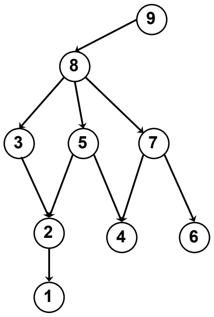
Let $\{0,1\}^*$ be the set of finite binary sequences, $\{0,1\}^\omega$ be the set of infinite binary sequences, and F be the set of sequences in $\{0,1\}^\omega$ that contain only a **f**inite number of occurrences of `1`'s.

(b) Describe a simple surjective function from $\{0,1\}^*$ to F.

(c) The set $\overline{F} ::= \{0,1\}^\omega - F$ consists of all the infinite binary sequences with *infinitely* many `1`'s. Use the previous problem parts to prove that $\overline{F}$ is uncountable.

Hint: We know that $\{0,1\}^*$ is countable and $\{0,1\}^\omega$ is not.

Problem 5 (Digraphs & Scheduling) (20 points).
The following DAG describes the prerequisites among tasks $\{1, \ldots, 9\}$.



(a) If each task takes unit time to complete, what is the minimum parallel time to complete all the tasks? Briefly explain.

(b) What is the minimum parallel time if no more than two tasks can be completed in parallel? Briefly explain.

Problem 6 (Directed Walks & Paths) (20 points).
An *Euler tour* of a graph is a closed walk that includes every edge exactly once.[2] In this problem we work out a proof of:

Theorem. *A connected graph has an Euler tour if and only if every vertex has even degree.*

(a) Show that if a graph has an Euler tour, then the degree of each of its vertices is even.

In the remaining parts, we'll work out the converse: if the degree of every vertex of a connected finite graph is even, then it has an Euler tour. To do this, let's define an Euler *walk* to be a walk that includes each edge *at most* once.

(b) Suppose that an Euler walk in a connected graph does not include every edge. Explain why there must be an unincluded edge that is incident to a vertex on the walk.

In the remaining parts, let **w** be the *longest* Euler walk in some finite, connected graph.

(c) Show that if **w** is a closed walk, then it must be an Euler tour.

Hint: part (b)

(d) Explain why all the edges incident to the end of **w** must already be in **w**.

(e) Show that if the end of **w** was not equal to the start of **w**, then the degree of the end would be odd.

Hint: part (d)

(f) Conclude that if every vertex of a finite, connected graph has even degree, then it has an Euler tour.

[2] In some other texts, this is called an *Euler circuit*.

Problem 7 (Partial Order & Equivalence) (20 points).
Say that vertices u, v in a digraph G are *mutually connected* and write

$$u \overset{*}{\longleftrightarrow} v,$$

when there is a path from u to v and also a path from v to u.

(a) Prove that $\overset{*}{\longleftrightarrow}$ is an equivalence relation on $V(G)$.

(b) The blocks of the equivalence relation $\overset{*}{\longleftrightarrow}$ are called the *strongly connected components* of G. Define a relation $\rightsquigarrow$ on the strongly connected components of G by the rule

$$C \rightsquigarrow D \quad \text{IFF there is a path from some vertex in } C \text{ to some vertex in } D.$$

Prove that $\rightsquigarrow$ is a weak partial order on the strongly connected components.

Problem 8 (**Simple Graphs: Degree & Isomorphism**) (**20 points**).

Circle true or false for following statements about trees and *provide counterexamples* for those that are false.

(a) Any connected subgraph is a tree. **true** **false**

(b) Adding an edge between two nonadjacent vertices creates a cycle. **true** **false**

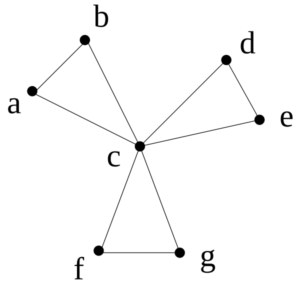
(c) The number of vertices is one less than twice the number of leaves. **true** **false**

(d) The number of leaves in a tree is not equal to the number of non-leaf vertices. **true** **false**

(e) Any subgraph of a tree is a tree. **true** **false**

(f) The number of vertices in a tree is one less than the number of edges. **true** **false**

Problem 9 (**Simple Graphs: Degree & Isomorphism**) (**20 points**).
Let G be the following graph:



(a) How many isomorphisms are there from G to itself? Show your work or we can't award any partial credit.

(b) Let $f : V(G) \to V(G)$ be a randomly chosen bijection from $V(G)$ to itself. What is the probability that f is an isomorphism given $f(a) = a$.

Problem 10 (**Bipartite Matching**) (**20 points**).

Scholars through the ages have identified *twenty* fundamental human virtues: honesty, generosity, loyalty, prudence, completing the weekly course reading-response, etc. At the beginning of the term, every student in Math for Computer Science possessed exactly *eight* of these virtues. Furthermore, every student was unique; that is, no two students possessed exactly the same set of virtues. The Math for Computer Science course staff must select *one* additional virtue to impart to each student by the end of the term. Prove that there is a way to select an additional virtue for each student so that every student is unique at the end of the term as well.

Suggestion: Use Hall's theorem. Try various interpretations for the vertices on the left and right sides of your bipartite graph.