CS 173 A, Fall 2018, Midterm 2 preparation

Tandy Warnow

November 12, 2018

1 Preparing for Midterm 2

1.1 Basic information

This midterm will have 6 problems, each worth 20 points - one with a proof (where you show all your work) and the other 5 being short problems (calculations or multiple choice) where you only show your answer. Your grade will be calculated by taking the 5 highest scoring problems for you and adding them up, and so will be out of 100. You can skip one problem if you wish or do them all, and we'll take the best 5.

As with the first midterm, it is closed book.

The conflict exam will take place in the ECE Grainger Auditorium during the regularly scheduled class period. You will need to make sure you go to the correct room – either DCL or Grainger – based on your discussion section; please make sure you know your discussion section and go to the correct room! Please also make sure you know your T.A.'s name, since you'll write it on the exam paper.

As with Midterm 1, if you wish a regrade on any problem, you will need to leave the midterm with the T.A. after you see it. However, the TAs will be doing the regrading this time.

One thing to realize is that all the material you learned up to Midterm 1 is also implicitly assumed for Midterm 2. That means, for example, that you know what binary relations are, set-builder notation, functions, etc. The midterm may have questions that draw on that material.

1.2 Proof

Unless you are asked to do your proof using a particular technique, you can do it however you like, as long as it's convincing. For example, you may want to do a direct proof, or a proof by contradiction, instead of a proof by induction. Or you may want to use induction! Remember also that you can use things you've been taught - such as the Pigeonhole Principle, and that every finite subset of the real numbers has a minimum as well as a maximum element.

There will be one problem involving a proof, from one of the three categories below:

- Suppose you have an oracle for some problem, which could be any kind of problem - such as a decision problem, an optimization problem, or a construction problem. Show that you can solve some other problem with a polynomial number of calls to the oracle plus a polynomial amount of other work. As examples:
  - think of how an oracle for the decision problem for SAT can be used to find a satisfying assignment (when it exists) for an input to SAT
  - think of how to use an oracle that tells you the minimum number of colors you need to properly vertex-color a graph in order to construct the minimum coloring.
  - or how to use an oracle that returns a maximum clique in order to find a maximum independent set
- Prove something about a graph using induction (either on the number of edges or the number of vertices). As an example,
  - recall how we showed you can prove that every simple graph has an even number of vertices of odd degree by doing induction on the number of edges in the graph.
  - recall how we showed that every tree on n vertices has exactly $n-1$ edges.
- Prove you can solve a problem in polynomial time for a particular class of graphs. For this, please look at the solutions to Homework 7.

Some advice about doing proofs

- First understand the question. Read the definitions of the terms used in the problem before you try to do the problem.
- Convince yourself that what you are asked to prove is true. Draw examples and try things out. Do lots of examples.
- Draw more pictures.
- Come up with an explanation in English (with pictures) of why the statement is true. Then turn it into a more formal argument.

Some specific examples are helpful.

Proof of a recent homework problem Suppose you wish to prove that $G = (V, E)$ has a clique of size k if and only if $H = (V', E')$ has a clique of size $k + 1$ where

- $V' = V \cup \{v\}$
- $E' = E \cup \{(a, v) | a \in V\}$

In other words, H is formed by adding a new vertex v and making v adjacent to every vertex in V.

Since you are asked to prove an *if-and-only-if* statement, you need to prove both directions. Do them both separately.

Proving that if G has a k-clique then H has a $(k+1)$-clique: We assume G is arbitrary and has a clique of size k. **Let's name the clique it has so we can work with it** – let's call it X.

Draw pictures of graphs G and H and try to see what the clique of size $k+1$ would be in H. What do you think? It will help if you make X a *largest* clique in G, and see what the $(k+1)$-clique in H will have to look like.

Do you see a pattern? Is there always a clique of size $k+1$ in H?

If you try enough graphs, you'll see that the set formed by adding v to X is a clique and it has $k+1$ vertices. So name this set $Y = X \cup \{v\}$.

Even if you don't know how to *prove* that Y is a clique, you have at least made the claim that Y is what you want. (And so you'll get partial credit.)

Then you need to finish the proof that Y is a clique. The proof that Y is a clique follows from X being a clique and v being adjacent to every vertex in G. So every pair of vertices a, b in Y are adjacent - either both are in X and so adjacent because X is a clique, or $a = v$ and so adjacent by construction of H. Hence Y is a clique and it has $k+1$ vertices.

Since G was arbitrary, the statement is true for all graphs G. Therefore you have shown one direction.

Proving that if H has a $(k+1)$-clique then G has a k-clique: Once again, we assume G is arbitrary, and we assume H has a $(k+1)$-clique. We name that $(k+1)$-clique Y so we can work with it.

Once again we draw pictures... but note that we cannot assume that Y includes vertex v. So we have two different types of pictures - ones where Y has v and the others where Y does not contain v. Obviously we have to do a case analysis.

- Case: $v \in Y$. For this case, consider $X = Y \setminus \{v\}$. Note that $X \subseteq V$. Also, since Y is a clique, X is also a clique, and so X is a k-clique in G.
- Case: $v \notin Y$. For this case, then Y is a subset of the vertices in G, and so G has a $(k+1)$-clique, and hence also a k-clique (take any subset of k vertices of Y and you're done.)

Thus, for both cases, if H has a $(k+1)$-clique then G has a k-cllique.

Since G was arbitrary, the statement is true for all graphs G. Thus we have proved the other direction.

1.3 Short problems

The short problems will cover the topics given below.

1. 2 problems will be about graphs.

- Graph terminology. Examples include: cycle, path, circuit, clique and maximum clique, independent set and maximum independent set, dominating set and minimum dominating set, vertex coloring (note it is assumed to be "proper") and minimum vertex coloring, matching and maximum matching, connected, adjacent, incident, component, etc. Know what a degree of a vertex is. Know also the associated terms of clique number, chromatic number, matching number, etc.). Know what the distance between two vertices in a graph is. Know what a Hamiltonian path, Eulerian walk, Eulerian circuit, and Hamiltonian cycle are. Know the common families of graphs (e.g., K_n, $K_{m,n}$, W_n, C_n, and P_n). Know what the complement of a graph is (recall you won't add in self-loops)
- Be able to reason about something, given its definition. For example, the "degree sequence" in a graph and the definition of a "regular graph", see Reading Quiz 9.
- Phrase real-world problems in terms of graphs
- Read mathematics and apply it to a graph
- Solving problems on graphs
- Know which graph problems are NP-hard, and which can be solved in polynomial time

2. Computational complexity

   - Know what a decision problem is
   - Know the definition of $\mathcal{NP}, \mathcal{P}$, and $Co - \mathcal{NP}$.
   - Know the required properties for a Karp reduction from problem π to problem π', and be able to describe one
   - Know the standard NP-hard problems (and recognize them when they appear in a different form)
   - Know what CNF means, and similarly what 2-SAT, 3-SAT, and SAT mean.
   - Know the relationship between decision, optimization, and construction problems, and how oracles for one can be used to solve the others

3. Big-O

   - Running time analysis of simple algorithms
   - Definitions of Big-O, and answering questions about Big-O (such as whether one function f is $O(g)$ where f and g are defined)
   - Be able to find the constants C and k to establish that f is $O(g)$, where f and g are defined.

4. Combinatorial counting

- Know the different techniques: (a) generating all objects at the leaves of a decision tree, and potentially correcting for over-counting, (b) dividing into disjoint cases and doing a case analysis, and (c) counting the complement
- Know some basic formula (like what $C(n, k)$ is – and by the way, $C(0, 0)$ is defined to be 1).
- Be able to do some straightforward counting

5. Reading and writing mathematics, especially in the context of graphs

2 Sample problems involving proofs

Proofs by induction:

1. Prove by induction on the number of edges that for every simple graph $G = (V, E)$, the number of vertices of odd degree is even.
2. Prove by induction on n that every tree with n vertices has $n - 1$ edges.
3. Prove using induction on n that any finite simple graph with exactly $2n$ vertices and no clique of size 3 has at most n^2 edges.

Proofs that some problems can be solved in polynomial time for special graphs:

1. Prove that the construction problem for minimum dominating set can be solved in polynomial time for C_n for all $n \geq 3$.
2. Prove that the construction problem for minimum vertex cover can be solved in polynomial time for W_n for all $n \geq 3$.

Proofs that you can use oracles for some problems to solve other problems

1. Suppose you have an oracle to solve the construction problem for Minimum Vertex Cover. How can you use it to solve the construction problem for Maximum Independent set? (And vice versa?)
2. Suppose you have an oracle for the decision problem SAT. Show that you can find a satisfying assignment for any input to SAT that is satisfiable using at most n calls to the oracle, where n is the number of variables in the input.
3. Suppose you have an oracle for the decision problem CLIQUE. Show that you can find the size of a maximum clique using at most n calls to the oracle, where n is the number of vertices in the input to CLIQUE. Then show you can find a maximum clique with at most n additional calls to the oracle.
4. Suppose you have an oracle for the decision problem CLIQUE. Show that you can solve the optimization problem MAXIMUM INDEPENDENT SET using a polynomial number of calls to the oracle and a polynomial amount of extra work.
5. Suppose you have an oracle for the decision problem INDEPENDENT SET. Show you can use it to solve the decision problem VERTEX COVER with a polynomial number of calls to the oracle.
6. Suppose you have an oracle for solving the decision problem MATCHING. Show that you can find the size of a maximum matching using at most n calls to the oracle, where n is the number of vertices in the input to MATCHING. Then show you can find a maximum matching with at most n additional calls to the oracle.

3 Sample "short" problems

Remember the short problems won't have any proofs, and you will not be asked to show any work. But to prepare for these, it can be helpful to try out problems where you do need to show your work. So some of the sample problems below are stated as"prove or disprove" something. It is helpful (for preparing for the final, or future homework) to try to do the proofs. By the way, a "disproof" means finding a counterexample. For example, if I ask you to prove or disprove "For all simple graphs $G = (V, E)$ with at least two nodes, there are two nodes with the same degree", you would either prove it true, or else show a graph where every two vertices have different degrees.

The reading quizzes are a good source of sample "short problems". Additional sample problems are:

3.1 Problems on graphs

1. Suppose you know all the people in the city of LAUGHING GAS, TEXAS. And therefore you know who knows who. You want to find a group of people so that everyone else in the city knows at least one person in that set. You also don't want to have that set be big - so the smallest set is best. Formulate this as a graph problem – what is the graph, and what graph problem do you need to solve?

2. True or false? (Try to prove your answer correct.) For every graph G, every vertex cover is a dominating set.

3. True or false? (Try to prove your answer correct.) For every simple graph $G = (V, E)$ with at least two vertices, there are two vertices with the same degree.

4. True or false? (Try to prove your answer correct.) For every graph G, every dominating set is a vertex cover.

5. True or false? (Try to prove your answer correct.) For every connected graph, every vertex cover is a dominating set.

6. Which of the following statements are true? Give a proof if you think it is true, otherwise show a counterexample.

   - If A is a vertex cover of graph G then A is dominating set of graph G.
   - If A is dominating set of graph G then A is a vertex cover of graph G.
   - Every tree is 2-colorable.
   - Every cycle is 2-colorable
   - Every cycle is 3-colorable
   - Every wheel graph is 3-colorable

- Every dominating set is a vertex cover in C_n, for all $n \geq 3$
- Every vertex cover is a dominating set in C_n, for all $n \geq 3$
- Every dominating set is a vertex cover in W_n, for all $n \geq 3$
- Every vertex cover is a dominating set in W_n, for all $n \geq 3$
- Every 2-colorable graph is bipartite
- Every 2-colorable graph is a tree
- For every graph G, its chromatic number is at least as big as its clique number
- If A is a vertex cover in a graph G, then the graph produced by removing the vertices of A from G has no edges.
- If A is a dominating set in a graph G, then the graph produced by removing the vertices of A from G has no edges.

7. Draw the complement graph (remember not to include self-loops) for each of the following graphs. Then, for each of the graphs and its complement, compute the chromatic number, maximum clique size, maximum independent set size, minimum vertex cover size, and minimum dominating set size. Also determine if the graph has an Eulerian circuit or Eulerian walk.

   - C_4
   - W_4
   - P_4

8. Calculate the number of edges for each of the following graphs, as a function of n: $P_n, C_n, K_n, K_{2n,n}$.

9. Let $G = (V, E)$ denote a given graph, and let $A \subseteq V$. Let $P(A, G)$ denote the assertion:

   $$\forall x \in V \setminus A, \exists y \in A \ s.t. \ (x, y) \notin E$$

   For each of the graphs G given below, give an example of a set A that satisfies $P(A, G)$. Then try to find the smallest set A that satisfies $P(A, G)$. Then, try to prove that your choice for A satisfies $P(A, G)$. Finally, prove that for all graphs G there is at least one set A that satisfies $P(A, G)$.

   - $G = K_n$
   - $G = C_n$
   - $G = W_n$
   - $G = K_{n,m}$
   - $G = P_n$

10. Draw a graph (not a big one, as that will be too difficult) and compute

    - Minimum vertex cover

- Maximum independent set
- Maximum clique
- whether it is 2-colorable
- whether it has an Eulerian circuit or Eulerian walk
- minimum dominating set
- maximum matching

3.2 Problems about complexity theory

Start by remembering what the sets $\mathcal{P}, \mathcal{NP}$, and $co-\mathcal{NP}$ refer to. Also know the terms $\mathcal{NP}$-hard and $\mathcal{NP}$-complete, and what the question "Does $\mathcal{P} = \mathcal{NP}$?" means. Remember also what a Karp reduction does, and *what* showing that $\pi \propto \pi'$ means. Make sure you understand the consequences that would follow if some problem that is $\mathcal{NP} - hard$ is also solvable in polynomial time. Think about how an optimization problem or construction problem can be $\mathcal{NP}$-hard, so that this is not only about problems that are *in* $\mathcal{NP}$.

For each question that follows, indicate if it is true or false.

1. If SAT (the decision problem) can be solved in polynomial time, then so can the decision problem 3-COLORABILITY.

2. The problem that takes as input a graph $G = (V, E)$ and returns a subset $X \subseteq V$ such that X is a vertex cover and has the smallest number of vertices is in $\mathcal{NP}$.

3. If the construction problem for MAX CLIQUE (i.e., finding the maximum clique) can be solved in polynomial time, then $\mathcal{P} = \mathcal{NP}$.

4. If the decision problem for SAT can be solved in polynomial time, then the construction problem (finding a satisfying assignment if one exists) can also be solved in polynomial time.

5. The decision problem 2-COLORABILITY is in $\mathcal{NP}$.

6. The decision problem 3-COLORABILITY is $\mathcal{NP}$-complete.

3.3 Counting problems:

Give the formulas for the following counting problems, as functions of n and k. Note: you can, and should, use the notation you've been taught. So you can write $C(x, y)$ instead of the formula for $C(x, y)$. You can write $x!$ instead of $x \times (x-1) \times (x-2) \times \ldots 2 \times 1$.

1. You have k boxes and n books, and you want to put all the books into boxes. How many ways can you assign books to boxes?

2. You have $2k$ friends and all of them are bored. You want to entertain them and take them out, but only two at a time. So each night for the next k nights you'll pick two friends and take them out. How many ways can you do this, if the choice of night matters? (Note that picking Sally and Henry for the same night is the same as picking Henry and Sally - order of the people within one evening is irrelevant.)

3. You have $2k$ friends and all of them are bored. And so bored that you have to do something right away. Instead of taking them out on successive nights, you are going to just treat them all to dinner, in pairs, all tomorrow night. What matters is only who gets paired with whom, and not where they go to dinner. How many ways can you do this?

4. You have a set of n different people, $\{p_1, p_2, \ldots, p_n\}$. In each case assume $1 \leq k \leq n$.

   - How many ways can you select k of these to a party, if order doesn't matter?
   - How many ways can you select k of these if p_1 *cannot* be one of the people selected (and order doesn't matter)?
   - How many ways can you pick people to go out with for the next k nights, one at a time and never repeating anyone twice? (Note that order matters.)
   - How many ways can you pick people to go out with for the next k nights, one at a time, but you can invite a person as often as you wish. (Note that order matters.)

3.4 Big-O problems:

1. Define what it means for f to be Big-O of g where f and g are both functions from positive integers to positive integers. (Be able to provide a definition in terms of the existence of two positive constants, C and k.)

2. Know some techniques to figure out whether a function is Big-O of another function.

3. For each pair of functions f and g given below (with both mapping natural numbers to natural numbers) say whether f is $O(g)$ and whether g is $O(f)$. Note that it is possible for both to be Big-O of each other or for neither to be Big-O of each other.

   (a) $f(n) = 1$ and $g(n) = 5$

   (b) $f(n) = n \log n$ and $g(n) = n^{1.5}$

   (c) $f(n) = 3^n$ and $g(n) = 2^{2n}$

   (d) $f(n) = n^3 + n^4$ and $g(n) = 5n^3$

   (e) $f(n) = 1000$ if $n < 50$ and $f(n) = n^2$ if $n \geq 50$, $g(n) = n$

(f) $f(n) = 2^n$, $g(n) = 2^{n+3}$

(g) $f(n) = 2^n$, $g(n) = 3^n$

(h) $f(n) = 2^n$, $g(n) = n^2 2^{n-1}$

(i) $f(n) = 2^{n \log_2 n}$, $g(n) = n^n$

(j) $f(n) = n^3 + 5000$, $g(n) = n^3 - 5000$

(k) $f(n) = 1$ if n is odd and $f(n) = n$ if n is even, and $g(n) = n$

(l) $f(n) = 1$ if n is odd and $f(n) = n$ if n is even, and $g(n) = 2n$ if n is odd and $g(n) = 5$ if n is even