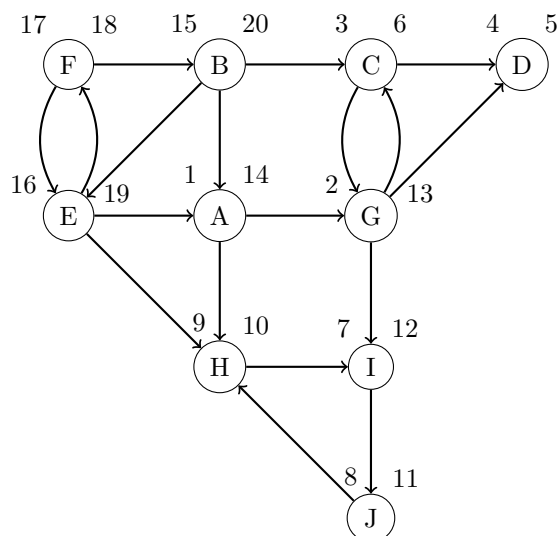


CS170 Midterm 1 – Solutions

Problem 1

1. (7 points)

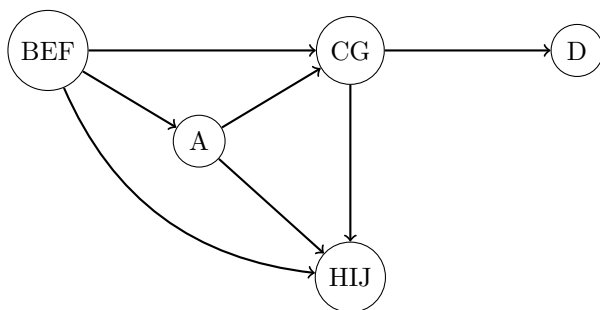


Pre order numbers are to the upper left of each node, post order numbers are to the right.

Cross edges are $B \rightarrow A$, $B \rightarrow C$, $E \rightarrow A$, and $E \rightarrow H$.

Grading: 5 points for correct pre and post order numbers. 2 points for identifying any cross edges. Points were deducted based on how many of the pre and post order numbers were wrong or for identifying incorrect cross edges or not identifying any at all.

2. (4 points)



Grading: 4 points for the correct SCC DAG. 1-2 points were deducted for wrong edges in the DAG based on how many edges were introduced or missed. 2 points were deducted for incorrectly identifying the SCCs by at most one node, but having the graph otherwise correct.

3. (4 points)

The minimum number of SCCs in the resulting graph is two. We can achieve this number by adding an edge from any of H, I, J or D to any of B, E, or F. The reason two is the minimum (and not one) is that there are two sink nodes in the SCC DAG. Adding any single edge could only eliminate at most one of the sink nodes.

Grading: 2 points for identifying two as the minimum number of nodes in the resulting SCC DAG. Note that if your graph was incorrect for part 2, we tried to be consistent with your answer there. 2 points for justifying why two was the minimum number of SCCs after adding a single edge. Simply stating what the edge is and showing that it reduces the graph to two SCCs does not tell us why we cannot reduce the graph to one SCC.

Problem 2

1. (1 of 5 points) $5^{-1} \cong 17 \pmod{21}$.

One way to compute this is to use the extended Euclidean algorithm.

$$21 = 4 \cdot 5 + 1$$

$$5 = 5 \cdot 1 + 0$$

We find $21 - 4 \cdot 5 = 1$, so $5^{-1} \cong -4 \cong 17 \pmod{21}$.

Grading: 1 point for a correct answer, or 1/2 point for a good attempt or indicating that one is trying to find x such that $5x \cong 1 \pmod{21}$.

2. (2 of 5 points) 12 of the 21 integers modulo 21 have inverses.

An integer has an inverse iff it is coprime to 21, iff it is divisible by neither 3 nor 7. Of the 21 integers, 7 are divisible by 3, 3 are divisible by 7, and one is divisible by both. $21 - 7 - 3 + 1 = 12$.

Grading: 2 points for a correct answer, or 1 for saying an integer has an inverse iff it is coprime to 21.

3. (2 of 5 points) $5^x \cong 1 \pmod{21}$ iff x is a multiple of 6.

This can be verified by trying the values $5^1, 5^2, \dots \pmod{21}$ and seeing that the first 1 appears when $x = 6$.

The easiest way to discover this is to use the Chinese Remainder Theorem: $5^x \cong 1 \pmod{21}$ iff $5^x \cong 1 \pmod{7}$ and $5^x \cong 1 \pmod{3}$. By Fermat's little theorem, if x is divisible by 6, then $5^x \cong 1 \pmod{7}$. Conversely, 5^2 and 5^3 are not $\cong 1 \pmod{7}$, so we know $5^x \cong 1 \pmod{7}$ iff x is a multiple of 6. In a similar way, we can determine that $5^x \cong 1 \pmod{3}$ iff x is even. By the Chinese Remainder Theorem, then, $5^x \cong 1 \pmod{21}$ iff x is a multiple of 6.

Grading: Two points for the correct answer. Otherwise, one point for noticing that $5^x \cong 1 \pmod{21}$ if x is a multiple of 12, or for stating other true facts about x , besides just saying $5^x \cong 1 \pmod{21}$.

4. (5 points) We make five calls to `divandconq(n/2)`, and then print n^2 lines. On input n , call the number of lines printed $T(n)$. Then we have the recurrence

$$T(n) = 5T(n/2) + n^2.$$

Using the Master Theorem, $T(n) = \Theta(n^{\log_2 5})$.

Grading: Two points for stating the recurrence relation. Three points for finding the correct solution, minus one point if the solution wasn't simplified.

5. (5 points)

- (2 of 5 points) ω is any primitive eighth root of unity: $\omega = e^{i2\pi/8} = \cos(\pi/4) + i \cdot \sin(\pi/4)$.

Grading: Two points for any primitive eighth root of unity written as $\omega = \pm \cos(\pi/4) \pm i \sin(\pi/4)$ or $\omega = \pm 1/\sqrt{2} \pm i \cdot 1/\sqrt{2}$ or similar. Otherwise, one point for writing $\omega = e^{i2\pi/n}$ or that ω is a $1/n$ rotation or similar.

- (3 of 5 points) $A_2 = A(\omega^2) = -1 - i$ or $-1 + i$ depending on which value for ω we choose.

Our polynomial is $A(x) = 1 - x + x^2 - x^3 + x^6 + x^7$. $\omega^2 = \pm i$; the easiest way to tell is that ω^2 must be a primitive fourth root of unity. So $A_2 = A(\omega^2) = A(i) = -1 - i$.

Grading: Three points for the correct answer, or for a wrong answer in $x + i \cdot y$ form as long as you wrote that $\omega^2 = \pm i$, followed some steps that should have worked but with an addition or multiplication error. Otherwise, two points for the right idea but not written in $x + i \cdot y$ form, or for evaluating $A(\omega^k)$ for some $k \neq 2$.

Problem 3

It is instructive to compare formal proofs with brief arguments given below, and observe how the brief arguments contain most of the ideas from and the general outline of the formal proofs (by mentioning key “players” of the proofs, e.g. what exactly is the contradiction, which and how the cycle is formed, the use of quantification “every”, etc). Note also that standard terminology (e.g. order preserving) are used whenever possible, to avoid potential confusion. It is perfectly okay (and usually recommended in tests) to write a bit more than the following briefest arguments, because often it makes your ideas clearer.

For grading, one point for correct *True/False*, and three points for explanations supporting the *True/False*. We tried to grade answers based on the ideas, but for consistency we need to base our grading on what was written on the midterm, rather than on subsequent clarifications of answers after midterm. That’s why it is important to try to be precise in giving answers, to make the ideas clearer.

(A) If all weights of a graph are distinct, the minimum spanning tree is unique: **True**.

Remark: We present two ideas. The first one gives a clean proof using the cut property, and is recommended. The second one gives a correct but less intuitive, “iterative” proof presented in Lecture.

Formal Proof 1: Fix a cut (S, \bar{S}) . The proof of the cut property shows that every MST must use some lightest edge crossing S and \bar{S} . When the edge weights are distinct, there is a unique lightest edge crossing S and \bar{S} , and so every MST must use this edge. It follows that MST is unique. (*A bit more verbosely, consider an MST \mathcal{T} . Consider any edge $e \in \mathcal{T}$, then $\mathcal{T} - e$ creates a cut. Now e must be the lightest edge crossing S and \bar{S} , so any MST must also use e .*)

Brief Argument 1: The lightest edge crossing any cut is unique, and must be in every MST by (the proof of) the cut property.

Formal Proof 2: Assume for contradiction that there are two different minimum spanning trees \mathcal{T} and \mathcal{T}' . So there must be some edge in exactly one of \mathcal{T} and \mathcal{T}' (but not both), let e be the heaviest one, and assume $e \in \mathcal{T}$ but $e \notin \mathcal{T}'$. Take away e from \mathcal{T} , and \mathcal{T} breaks into two connected components S and \bar{S} . In \mathcal{T}' , there must be at least one edge e' crossing the cut between S and \bar{S} , where $e' \neq e$. Since \mathcal{T} is a MST, we conclude that $\text{cost}(e) \leq \text{cost}(e')$, for otherwise the tree $\mathcal{T}'' = \mathcal{T} - e + e'$ would have cost less than \mathcal{T} (*this is essentially the proof of the cut property*). Distinctness of edge weights gives $\text{cost}(e) < \text{cost}(e')$, but e' is also in exactly one of \mathcal{T} and \mathcal{T}' , contradicting our choice for e (as the heaviest such edge).

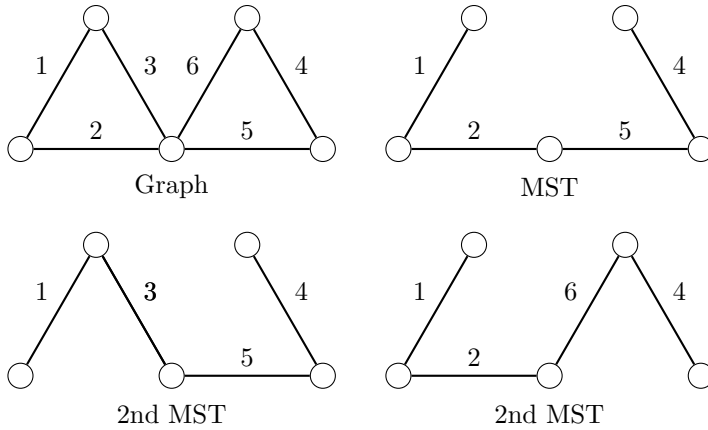
Brief Argument 2: Consider the two MSTs \mathcal{T} and \mathcal{T}' . Whenever \mathcal{T} has an edge e that is not in \mathcal{T}' , \mathcal{T}' has a (strictly) heavier edge e' that is not in \mathcal{T} , where e' crosses the cut created by $\mathcal{T} - e$. Iterating this argument gives an infinite sequence of edges.

Remarks:

- *In both proofs, the main idea is to modify the proof of the cut property given in the textbook. Points for explanations were not awarded if the (cut/cycle) structure is not implied in the answers.*
- *Many answers mentioned that the output of a greedy MST algorithm (e.g. Kruskal’s, Prim’s) is unique. This statement, while true, is not sufficient to show the uniqueness of the minimum spanning tree, because those algorithms are not the only ways to find MSTs, a priori.*
- *Some answers seemed to state that if there are a list of distinct numbers, then no two equal-length sublists of them can have the same sum. This is not true.*
- *The grading is based on whether and how clear the cut property is modified to lead to a valid argument.*

(B) If all weights of a graph are distinct, the second shortest spanning tree is unique. **False**.

The following graph has two different second MST.



Remarks:

- Some answers seemed to have confused shortest paths with spanning trees, and got points deducted.
- Some answers mentioned and argued using a directed variant of MST, which is incompatible with what is taught in class, and got points deducted.

(C) If all weights of a graph are distinct, the second shortest spanning tree contains the shortest edge.
True.

Formal Proof: Assume not, there exists a second MST \mathcal{T} not containing the lightest edge (u, v) . Consider the path \mathcal{P} from u to v on \mathcal{T} , so \mathcal{P} and (u, v) form a cycle. For every edge e on \mathcal{P} , the spanning tree $\mathcal{T}_e = \mathcal{T} - e + (u, v)$ is lighter than \mathcal{T} , since (u, v) is the unique lightest edge. \mathcal{P} has at least two edges, so there are at least two different spanning trees lighter than \mathcal{T} , contradicting that \mathcal{T} is a second MST.

Brief Argument: The second MST \mathcal{T} and the lightest edge e form a cycle of length at least three, where by swapping e with edges of \mathcal{T} on the cycle, we get (at least) two spanning trees lighter than \mathcal{T} .

Remarks:

- Some answers mentioned swapping the “heaviest edge” with the “lightest edge” for lighter spanning trees. Without further quantification, this cannot be done, because the heaviest edge may be in every spanning tree.
- Some answers mentioned “it is (always) possible to find shorter spanning tree than the second MST”, which is not sufficient for a contradiction.

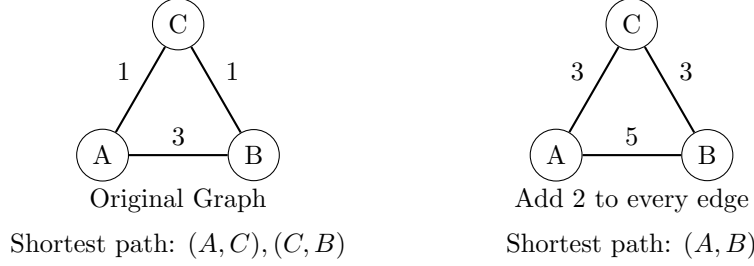
(D) If we add the same number to all weights of a graph, the minimum spanning tree remains the same.
True.

Brief Argument: Every spanning tree has exactly $|V| - 1$ edges, so after adding k to all weights, every MST gets its weight increased by $k(|V| - 1)$, and the ordering of spanning trees is preserved.

Remarks:

- A formal, one-line argument to justify the order-preserving property is as follows: $a < b$ iff $a + c < b + c$ for any real numbers a, b, c . Some answers mentioned this, making the idea clearer.
- Some answers just mentioned that the ordering of the edge weights are preserved. While true, this is insufficient to show that the MST is the same, strictly speaking.

(E) If we add the same number to all weights of a graph, the shortest path from A to B remains the same.
False.



Remarks:

- Some answers essentially gave a family of well-defined counter-examples, and were also accepted.

Problem 4

Let T_0 be the total sum of all the times for all the edges in the graph. In other words, $T_0 = \sum_{e \in E} T(e)$. For every edge (u, v) in the graph, consider a single weight function $w(u, v) = E(u, v) + (T(u, v)/100T_0)$. Running Dijkstra's algorithms with this single weight function sufficed to get the shortest path according to energy and if there were multiple shortest paths according to energy, then this one gives the shortest one prioritized according to time.

Proof sketch: Consider two paths P_1 and P_2 such that P_1 is more energy efficient. This implies $\sum_{(u,v) \in P_1} E(u, v) < \sum_{(u,v) \in P_2} E(u, v)$. Because all the energies are integers, $\sum_{(u,v) \in P_1} E(u, v) + 1/100 < \sum_{(u,v) \in P_2} E(u, v)$.

Now, consider the efficiency of P_1 and P_2 according to w . Note that $\sum_{(u,v) \in P_2} w(u, v) \geq \sum_{(u,v) \in P_2} E(u, v)$. Similarly, $\sum_{(u,v) \in P_1} w(u, v) = \sum_{(u,v) \in P_1} E(u, v) + (\sum_{(u,v) \in P_1} T(u, v))/100T_0 \leq \sum_{(u,v) \in P_1} E(u, v) + 1/100$.

Thus, $\sum_{(u,v) \in P_1} w(u, v) < \sum_{(u,v) \in P_2} w(u, v)$. Hence, paths maintain their ordering w.r.t. energy efficiency even under the new weighting function. Further, it is obvious that in case you have 2 paths with the same energy efficiency, $w(\cdot, \cdot)$ assigns less weight to the one which consumes less time.