Chapter 20 Practice Exercises

Disclaimer: These practice questions are not official and may not have been vetted for course material. You may use these for practice, but you should prioritize official resources from current staff for a more accurate depiction of what you need to know for assignments and exams.

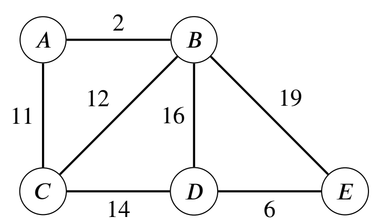
1. Which of the following statements must be **TRUE** regarding the minimum spanning tree (MST) of a graph with $|V|$ vertices?
   I. The MST is acyclic.
   II. The MST has $|V|$ edges.
   III. The MST cannot include the edge with the largest weight in the original graph.

   A) I only
   B) II only
   C) I and II only
   D) II and III only
   E) I, II, and III

2. Given a sparse graph with $|V|$ vertices and $|E|$ edges that is represented using an adjacency list, what is the time complexity of running the heap implementation of Prim's algorithm on this graph, provided that you use a binary heap as the underlying structure of your heap?
   A) $\Theta(|V|+|E|)$
   B) $\Theta(|E|\log(|V|))$
   C) $\Theta(|V||E|\log(|V|))$
   D) $\Theta(|V|^2)$
   E) $\Theta(|V|^3)$

3. Which of the following statements is **FALSE**?
   A) The time complexity of running Kruskal's algorithm on a graph with $|E|$ edges is $\Theta(|E|\log(|E|))$
   B) In terms of time complexity, the sorting algorithm involved in Kruskal's algorithm is the bottleneck of the entire algorithm
   C) The efficiency of Kruskal's algorithm relies on the efficiency of the union-find data structure
   D) For certain graphs, Kruskal's and Prim's algorithms may produce different MSTs
   E) Kruskal's algorithm does not work for graphs with negative edge weights

For questions 4-5, consider the following graph:



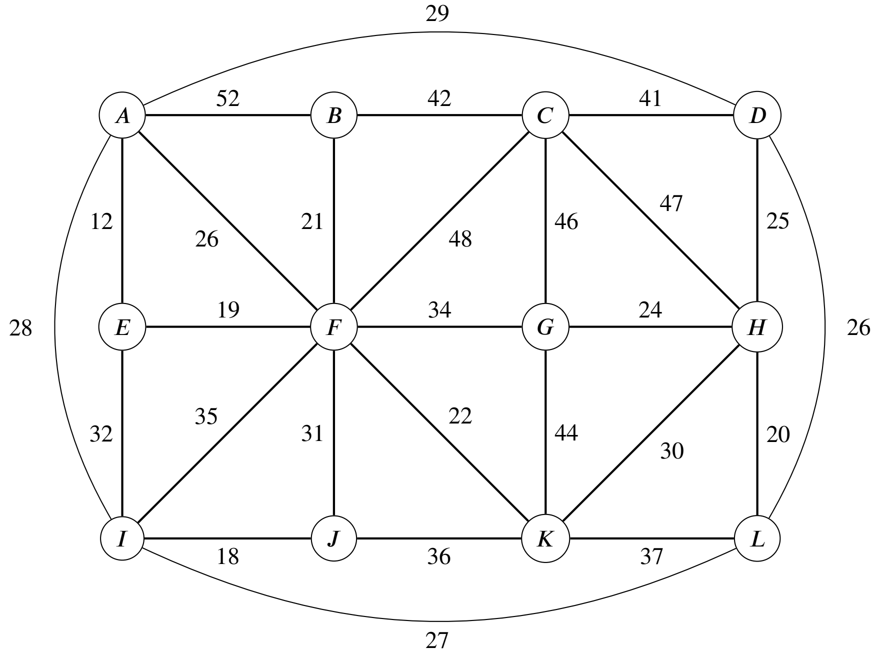
4. Using Prim's algorithm on the graph above (starting at vertex A), which vertex is added last?
   A) Vertex A
   B) Vertex B
   C) Vertex C
   D) Vertex D
   E) Vertex E

5. Using Prim's algorithm on the graph above (starting at vertex A), what is the total weight of the MST?
   A) 31
   B) 33
   C) 45
   D) 80
   E) None of the above

6. Given a graph and its MST, consider the following four scenarios:
   I. You increase the weight of an edge that is in the MST
   II. You decrease the weight of an edge that is in the MST
   III. You increase the weight of an edge that is not in the MST
   IV. You decrease the weight of an edge that is not in the MST

   For which of these scenarios could the MST change?
   A) I and III only
   B) I and IV only
   C) II and III only
   D) II and IV only
   E) I, II, III, and IV

7. Consider the following graph:



How many edges does this graph's MST have?
A) 10
B) 11
C) 12
D) 13
E) More than 13

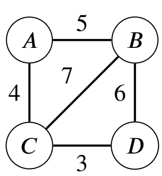
8. Suppose you are given the following distance matrix for five different vertices.

	$\mathbf{V_1}$	$\mathbf{V_2}$	$\mathbf{V_3}$	$\mathbf{V_4}$	$\mathbf{V_5}$
$\mathbf{V_1}$	-	9	12	10	7
$\mathbf{V_2}$	9	-	14	6	8
$\mathbf{V_3}$	12	14	-	13	11
$\mathbf{V_4}$	10	6	13	-	5
$\mathbf{V_5}$	7	8	11	5	-

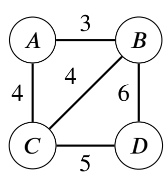
What is the total weight of the MST that connects these five vertices?
A) 26
B) 27
C) 28
D) 29
E) None of the above

9. For which of the following graphs is it possible for Prim's and Kruskal's algorithms to return different MSTs? *Select all that apply.*

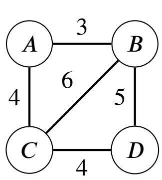
A)



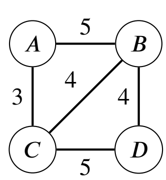
C)



B)



D)



E) None of the above

10. Suppose you are given the following distance matrix for five different vertices.

	V_1	V_2	V_3	V_4	V_5
V_1	-	21	22	14	17
V_2	21	-	11	20	13
V_3	22	11	-	18	15
V_4	14	20	18	-	19
V_5	17	13	15	19	-

If you run Prim's algorithm on this graph, starting from vertex V_1, what is the order in which the edges of the graph are added to the MST?
A) $V_1 - V_4$, $V_1 - V_5$, $V_2 - V_5$, $V_2 - V_3$
B) $V_1 - V_4$, $V_3 - V_4$, $V_2 - V_3$, $V_2 - V_5$
C) $V_1 - V_4$, $V_2 - V_3$, $V_2 - V_5$, $V_1 - V_5$
D) $V_1 - V_5$, $V_2 - V_5$, $V_2 - V_3$, $V_1 - V_4$
E) None of the above

11. What is the total weight of the MST for the graph provided in question 10?
A) 53
B) 54
C) 55
D) 56
E) None of the above

12. Suppose you are given the following distance matrix for six different vertices.

	V_1	V_2	V_3	V_4	V_5	V_6
V_1	-	12	22	17	21	10
V_2	12	-	18	20	19	11
V_3	22	18	-	15	14	24
V_4	17	20	15	-	13	23
V_5	21	19	14	13	-	16
V_6	10	11	24	23	16	-

If you run Kruskal's algorithm on this graph, in what order are the edges added to the MST?
A) $V_1 - V_6$, $V_2 - V_6$, $V_1 - V_2$, $V_4 - V_5$, $V_3 - V_4$
B) $V_1 - V_6$, $V_2 - V_6$, $V_4 - V_5$, $V_3 - V_5$, $V_1 - V_4$
C) $V_1 - V_6$, $V_2 - V_6$, $V_4 - V_5$, $V_3 - V_5$, $V_5 - V_6$
D) $V_1 - V_6$, $V_2 - V_6$, $V_3 - V_5$, $V_3 - V_4$, $V_5 - V_6$
E) None of the above

13. What is the total weight of the MST for the graph provided in question 12?
A) 60
B) 64
C) 65
D) 66
E) None of the above

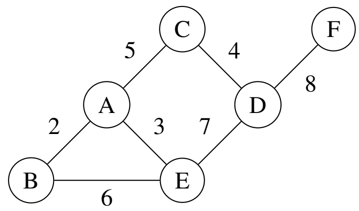
14. You are given a simple, connected graph G with 30 vertices. All of the edges in G initially have a weight of 1. Suppose you randomly select 20 edges in G and cut each of their weights in half. After doing so, what is the lowest possible weight a spanning tree in G can have?
A) 9
B) 10
C) 19
D) 20
E) None of the above

15. Suppose you are given two weighted graphs A and B with identical vertices and edges, but with different edge weights. You are given another weighted graph C, which also has identical vertices and edges, but with edge weights equal to the sum of A and B. Let's denote the total weights of the MSTs of A, B, and C as MST_A, MST_B, and MST_C. Which of the following expressions could potentially be true?
I. $MST_A + MST_B > MST_C$
II. $MST_A + MST_B = MST_C$
III. $MST_A + MST_B < MST_C$

A) II only
B) I and II only
C) I and III only
D) II and III only
E) I, II, and III

Chapter 20 Exercise Solutions

1. **The correct answer is (A).** By definition, minimum spanning trees are acyclic, so statement I is true. Statement II is false because a MST has $|V|-1$ edges, not $|V|$ edges. Statement III is false because it is possible for the largest weighted edge in the graph to connect a vertex that cannot be reached from anywhere else, which necessitates it to be in the MST (also if every edge has the same weight). One example that disproves statement III is shown below (the edge from D to F of weight 8 must be in the MST):



2. **The correct answer is (B).** The time complexity of Prim's algorithm using a min-binary heap on an adjacency list is $\Theta(|E|\log(|V|)$ (see the section on Prim's algorithm for a detailed explanation).

3. **The correct answer is (E).** Kruskal's algorithm greedily selects the edges with the lowest weight, so negative weights are not a problem. Choice (A) is true because sorting the edges dominates with $\Theta(|E|\log(|E|))$ time. Choice (B) is true since sorting is indeed the bottleneck of the algorithm (adding edges and testing for cycles both take less than $\Theta(|E|\log(|E|))$ time). Choice (C) is true because Kruskal's relies on the union-find data structure to determine whether an edge can be added to the MST without introducing a cycle. Choice (D) is true because, if multiple MSTs exist for the same graph, the two algorithms may end up discovering different MSTs depending on their implementations (e.g., which vertex Prim's algorithm is initiated on, how equal weights are tiebroken in Kruskal's algorithm, etc.).

4. **The correct answer is (E).** Vertex E is added last. The process is shown below:

v	*k*	*d*	*p*
A	**T**	**0**	**-**
B	F	∞	-
C	F	∞	-
D	F	∞	-
E	F	∞	-

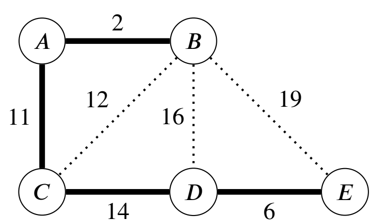
v	*k*	*d*	*p*
A	T	0	-
B	**T**	**2**	**A**
C	F	11	A
D	F	∞	-
E	F	∞	-

v	*k*	*d*	*p*
A	T	0	-
B	T	2	A
C	**T**	**11**	**A**
D	F	16	B
E	F	19	B

v	*k*	*d*	*p*
A	T	0	-
B	T	2	A
C	T	11	A
D	**T**	**14**	**C**
E	F	19	B

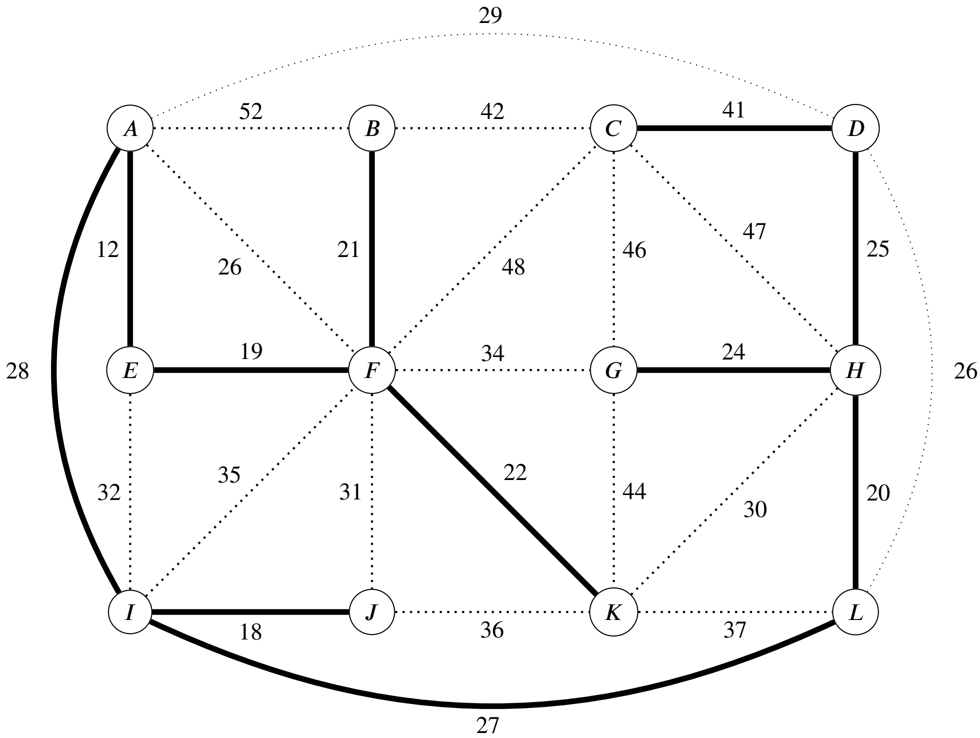
v	*k*	*d*	*p*
A	T	0	-
B	T	2	A
C	T	11	A
D	T	14	C
E	**T**	**6**	**D**

5. **The correct answer is (B).** The total weight of the MST is 2 + 11 + 14 + 6 = 33.

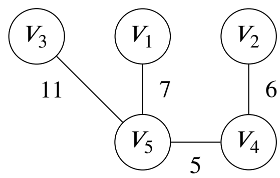


6. **The correct answer is (B).** In scenario I, increasing the weight of an edge in the MST makes the MST worse, so the MST may change if there is another edge that can be used to replace the edge that was modified. In scenario II, decreasing the weight of an edge in the MST only makes the MST better, so the MST would not change. In scenario III, increasing the weight of an edge outside the MST makes the alternatives worse, so the MST would not change. In scenario IV, decreasing the weight of an edge outside the MST makes the alternatives better, so the MST may change.

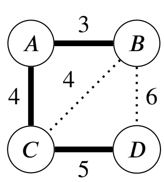
7. **The correct answer is (B).** A graph with $|V|$ vertices has $|V|-1$ edges in its MST, which is 11 in this case. For those adventurous enough to compute the MST of this graph, this is what the MST looks like:

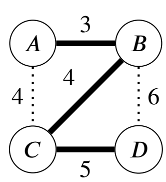


8. **The correct answer is (D).** The simplest strategy to solve this by hand is to use Kruskal's algorithm. Pick the smallest edge, check for cycles, and add to a running total if no cycles are created. First, we select the shortest edge of 5 between V_4 and V_5 and add it to the MST. Then, we look at the next shortest edge of 6 between V_2 and V_4. No cycle is involved, so we add this edge to the MST for a running total of $5 + 6 = 11$. Next, we look at the edge with a weight of 7, which connects V_1 and V_5. Again, no cycle is involved, so we add this edge to the MST for a running total of $11 + 7 = 18$. Next, we look at the edge with a weight of 8, which connects V_2 and V_5. Adding this edge would produce the cycle $V_2 \to V_4 \to V_5 \to V_2$, so we ignore it. Next, we look at the edge with a weight of 9, which connects V_1 and V_2. Adding this edge would produce the cycle $V_1 \to V_2 \to V_4 \to V_5 \to V_1$, so we ignore it. Next, we look at the edge with weight 10, which connects V_1 and V_4. Adding this edge would produce the cycle $V_1 \to V_4 \to V_5 \to V_1$, so we ignore it. Next, we look at the edge with weight 11, which connects V_3 and V_5. No cycle is involved, so we add this edge to the MST for a running total of $18 + 11 = 29$. Since all vertices are now in the MST, we are done. The final tree is shown below:



9. **The correct answer is (C).** Only this graph has two potential MSTs (the other three graphs have a single unique MST).





10. **The correct answer is (A).** We start at V_1 and add the external vertex with the shortest connection to V_1, which is V_4 with a weight of 14. Then, we add the external vertex with the shortest connection to V_1, V_4, which is V_5 with a weight of 17 to V_1. Then, we add the external vertex with the shortest connection to V_1, V_4, V_5, which is vertex V_2 with a weight of 13 to V_5. Then, we add the external vertex with the shortest connection to V_1, V_2, V_4, V_5, which is vertex V_3 with a weight of 11 to V_2.

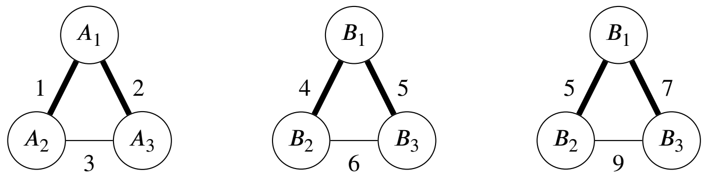
11. **The correct answer is (C).** The total weight of the MST is $11 + 13 + 14 + 17 = 55$.

12. **The correct answer is (C).** The edges in sorted order are 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22. We first add 10 ($V_1 - V_6$) and 11 ($V_2 - V_6$). We cannot add 12 ($V_1 - V_2$) because it would cause a cycle $V_1 \to V_2 \to V_6$. Next, we add 13 ($V_4 - V_5$) and 14 ($V_3 - V_5$). We cannot add 15 ($V_3 - V_4$) because it would cause a cycle $V_3 \to V_4 \to V_5$. Next, we add 16 ($V_5 - V_6$), and our MST is complete.

13. **The correct answer is (B).** The total weight of the MST is $10 + 11 + 13 + 14 + 16 = 64$.

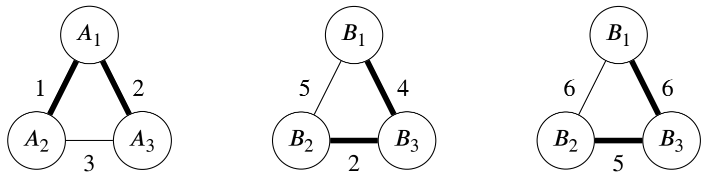
14. **The correct answer is (C).** A spanning tree of the original graph G must have 29 edges since there are 30 vertices. Without any modifications, the total weight of the spanning tree must also be 29, since every edge has a weight of 1. If we select 20 of these edges and cut their weights in half from 1 to 0.5, we reduce the total weight of the MST by $20 \times 0.5 = 10$. This yields a new minimum possible weight of 29 - 10 = 19.

15. **The correct answer is (D).** The case where $MST_A + MST_B = MST_C$ can occur when A and B share the same edges in the MST, which means that C must also have the same MST with a total weight that is the sum of the MSTs of A and B. An example is shown below:



$MST_A = 3$
$MST_B = 9$
$MST_C = 12$

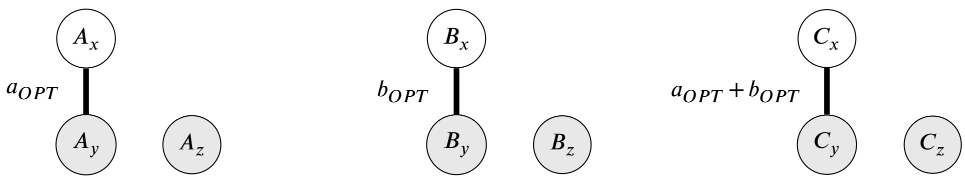
The case where $MST_A + MST_B < MST_C$ can occur when A and B share different edges in the MST. One example is shown below:



$MST_A = 3$
$MST_B = 6$
$MST_C = 11$

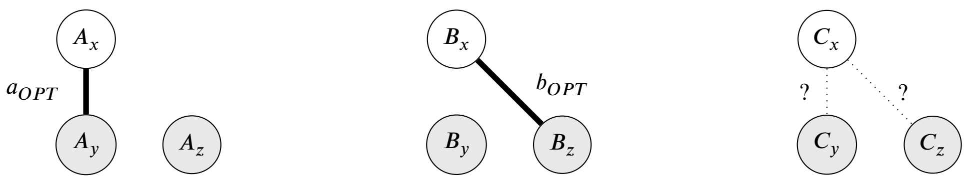
The case where $MST_A + MST_B > MST_C$ is a bit more interesting, since this implies that you can sum the edges of A and B to obtain a graph whose MST is better than the sums of the MSTs of A and B. Let's suppose that some arbitrary vertex A_x in graph A is initially connected to some other arbitrary vertex A_y in the MST. There are two possibilities here: the corresponding vertex B_x in graph B could also be connected to B_y in the MST, or it could be connected to some other arbitrary vertex B_z in the MST.

Scenario 1: B_x is initially connected to the MST of B using a matching edge.

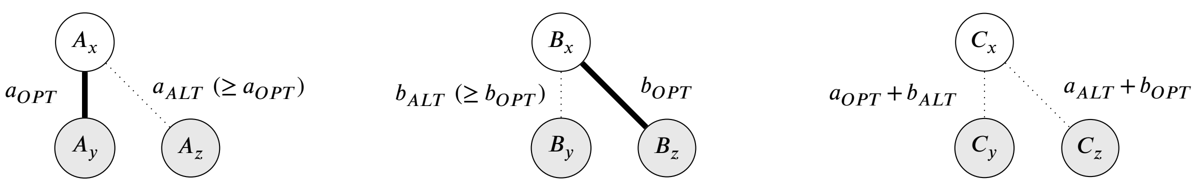


As shown above, when a vertex is connected to the MST using the same edge in both graphs A and B, we know that the cost to connect C_x to the MST of graph C must be $a_{OPT} + b_{OPT}$ (the shaded vertices are already part of the MST, i.e., "innies" using the terminology in section 20.2). If all MST edges match between A and B, then we get the case where $MST_A + MST_B = MST_C$.

Scenario 2: B_x is initially connected to the MST of B using the some other non-matching edge.



In this case, B_x in graph B is not connected to B_y in the MST, but instead through some other arbitrary vertex B_z. What happens now to the cost of adding C_x to the MST of C? In turns out that the cost of adding C_x to the MST of C in this scenario *cannot be better* than the cost when A_x and B_x are connected to the MST using the same matching edge. We know that the edge $C_x - C_y$ or the edge $C_x - C_z$ must be in the MST of C, since any other edge would imply that $A_x - A_y$ or $B_x - B_z$ would not be the cheapest way to connect A_x and B_x to the MSTs of A and B, respectively. However, we also know that the edges $A_x - A_z$ and $B_x - B_y$ (denoted as a_{ALT} and b_{ALT}) must have a weight that is no better than a_{OPT} and b_{OPT}, since they are not in the MST. As a result, both edges that can be used to connect C_x to the MST of C are either equal or worse off when compared to the scenario where the optimal edges of the MSTs were aligned.



This brings us to the following conclusion: if all the MST edges match between A and B, then we get the case where $MST_A + MST_B = MST_C$ (statement II). If there are any mismatching edges, then we could get the case where $MST_A + MST_B < MST_C$, since the MST of the combined graph C might be summing up and including inferior edge weights that are not in the MSTs of either A or B (statement III). However, there is no case where the combined graph may yield a better MST than the sums of the MSTs of A and B, since MST_C equaling $MST_A + MST_B$ occurs in the best case where all the optimal edges align. Thus, only statements II and III can potentially be true.