CS/ECE 374 P28

Abhay Varmaraja, Pengxu Zheng, Jiawei Tang

TOTAL POINTS

95 / 100

QUESTION 1

1 28.A. 15 / 20

- **- 0 pts** Correct
- ✓ **- 5 pts** **Minor error in ordering**
- **- 10 pts** Incorrect cost
- **- 20 pts** Incorrect ordering
- **- 15 pts** IDK

Minor error: you can only deploy a server at one location for one iteration, not two as your solution states. You will not have log(n) iterations - you still have a place a server at each location, so you will have n iterations. But the COST will come out to be O(nlogn).

QUESTION 2

2 28.B. 20 / 20

- **- 0 pts** Correct
- ✓ **- 0 pts** **Missing or incorrect proof that w(T) is a lower bound on deployment cost**
- **- 15 pts** Missing or incorrect proof that a closed walk under 2*w(T) exists
- **- 20 pts** Completely incorrect, low-effort proof
- **- 15 pts** IDK

QUESTION 3

3 28.C. 20 / 20

- ✓ **- 0 pts** **Correct**
- **- 5 pts** Minor error in proof
- **- 20 pts** Incorrect proof
- **- 15 pts** IDK

QUESTION 4

4 28.D. 40 / 40

- ✓ **- 0 pts** **Correct**
- **- 10 pts** Minor error in proof
- **- 20 pts** Incorrect or unclear cost calculation to get O(w(T) log n)
- **- 40 pts** Completely incorrect, low-effort proof.
- **- 30 pts** IDK

HW Solution

CS/ECE 374: Algorithms & Models of Computation, Spring 2019 Version: **1.0**

Submitted by:

- **≪Jiawei Tang≫**: **≪jiaweit2≫**
- **≪Pengxu Zheng≫**: **≪pzheng5≫**
- **≪Abhay Varmaraja≫**: **≪abhaymv2≫**

28

Solution:

28.A.

What we have here is a line-shape graph that every two vertices are connected with one edge. The ordering algorithm would be first we deploy the servers at the left-most and right-most vertices. Then we perform something like binary search where we deploy each server at the middle point. For example, we would deploy a server at the $\frac{n}{2}$ vertex for the first iteration. For the second, we would deploy servers at $\frac{n}{4}$ and $\frac{3n}{4}$ vertices, etc. We will have $log(n)$ iterations and each iteration will have cost $\frac{n}{2}$. Therefore, the total cost of deployment is $\Omega(nlogn)$.

28.B.

First, to prove that there exists a closed walk that visits all of vertices, we can use direct proof. Since now we know we have a minimum spanning tree T for the undirected graph G. Then we can be sure that there is a vertex v in G such that there is path from v to every other vertices. Because G is undirected, then there must exist a closed walk from v to every other vertices and when we are at the other vertices, we use the same edge to travel back to v. This closed walk will be consist of edges only in T if every other edges that are not in T have ∞ weight. Therefore, this closed walk will have $2w(T)$ weights and we can conclude that the total weight of edges of the walk is always at most $2w(T)$.

28.C.

Assume that there are at least two vertices in X. The closest pair of vertices, x, y, in X should be connected with each other with a single edge. Therefore, $d(x, y) = min(E)$ where E is the set of all edges in the graph. In a situation where all the vertices in X forms a cycle. Also because we have previously in part b, proved that there must be a closed walk visiting all vertices in X, we can say that the lower bound of weights of any closed walk in X is $d(x, y) \cdot |X|$. Also, we know from part b that the total weight of edges of a closed walk is at most $2w(T)$. Then we can conclude that

$$d(x, y) \cdot |X| \leq 2w(T)$$

28.D.(Discussed with Hengzhi's group)

By 28.C, we have proved that for the closest pair x,y of vertices in X, $d(x, y) \leq 2w(T)/|X|$. For each time the recursive greedy algorithm runs, there will be one less closest pair of vertices in X but the inequality above still holds. Then the total cost would be $\sum_{i=1}^{|X|} d(x_i, y_i)$. Because of the conclusion in 28.C, This total cost will be less than or equal to $\sum_{i=1}^{|X|} \frac{2w(T)}{|X|-i}$. We can see this sum is of harmonic series form as the $|X|-i$ can be replaced with i. By Jeff's notes(http://jeffe.cs.illinois.edu/teaching/algorithms/notes/99-recurrences.pdf) regarding harmonic series,

$$\sum_{i=1}^{|X|} \frac{2w(T)}{i} = 2w(T)H_{|X|} = \Theta(2w(T)log(|X|)) = \Theta(w(T)log(|X|))$$

Thus, the deployment cost is bounded by $O(w(T)log(|X|))$

1 28.A. **15** / **20**

- **- 0 pts** Correct
- ✓ **- 5 pts Minor error in ordering**
- **- 10 pts** Incorrect cost
- **- 20 pts** Incorrect ordering
- **- 15 pts** IDK

Minor error: you can only deploy a server at one location for one iteration, not two as your solution states. You will not have log(n) iterations - you still have a place a server at each location, so you will have n iterations. But the COST will come out to be O(nlogn).

HW Solution

CS/ECE 374: Algorithms & Models of Computation, Spring 2019 Version: **1.0**

Submitted by:

- **≪Jiawei Tang≫**: **≪jiaweit2≫**
- **≪Pengxu Zheng≫**: **≪pzheng5≫**
- **≪Abhay Varmaraja≫**: **≪abhaymv2≫**

28

Solution:

28.A.

What we have here is a line-shape graph that every two vertices are connected with one edge. The ordering algorithm would be first we deploy the servers at the left-most and right-most vertices. Then we perform something like binary search where we deploy each server at the middle point. For example, we would deploy a server at the $\frac{n}{2}$ vertex for the first iteration. For the second, we would deploy servers at $\frac{n}{4}$ and $\frac{3n}{4}$ vertices, etc. We will have $log(n)$ iterations and each iteration will have cost $\frac{n}{2}$. Therefore, the total cost of deployment is $\Omega(nlogn)$.

28.B.

First, to prove that there exists a closed walk that visits all of vertices, we can use direct proof. Since now we know we have a minimum spanning tree T for the undirected graph G. Then we can be sure that there is a vertex v in G such that there is path from v to every other vertices. Because G is undirected, then there must exist a closed walk from v to every other vertices and when we are at the other vertices, we use the same edge to travel back to v. This closed walk will be consist of edges only in T if every other edges that are not in T have ∞ weight. Therefore, this closed walk will have $2w(T)$ weights and we can conclude that the total weight of edges of the walk is always at most $2w(T)$.

28.C.

Assume that there are at least two vertices in X. The closest pair of vertices, x, y, in X should be connected with each other with a single edge. Therefore, $d(x, y) = min(E)$ where E is the set of all edges in the graph. In a situation where all the vertices in X forms a cycle. Also because we have previously in part b, proved that there must be a closed walk visiting all vertices in X, we can say that the lower bound of weights of any closed walk in X is $d(x, y) \cdot |X|$. Also, we know from part b that the total weight of edges of a closed walk is at most $2w(T)$. Then we can conclude that

$$d(x, y) \cdot |X| \leq 2w(T)$$

28.D.(Discussed with Hengzhi's group)

By 28.C, we have proved that for the closest pair x,y of vertices in X, $d(x, y) \leq 2w(T)/|X|$. For each time the recursive greedy algorithm runs, there will be one less closest pair of vertices in X but the inequality above still holds. Then the total cost would be $\sum_{i=1}^{|X|} d(x_i, y_i)$. Because of the conclusion in 28.C, This total cost will be less than or equal to $\sum_{i=1}^{|X|} \frac{2w(T)}{|X|-i}$. We can see this sum is of harmonic series form as the $|X|-i$ can be replaced with i. By Jeff's notes(http://jeffe.cs.illinois.edu/teaching/algorithms/notes/99-recurrences.pdf) regarding harmonic series,

$$\sum_{i=1}^{|X|} \frac{2w(T)}{i} = 2w(T)H_{|X|} = \Theta(2w(T)log(|X|)) = \Theta(w(T)log(|X|))$$

Thus, the deployment cost is bounded by $O(w(T)log(|X|))$

2 28.B. 20 / 20

- **- 0 pts** Correct
- ✓ **- 0 pts** **Missing or incorrect proof that w(T) is a lower bound on deployment cost**
- **- 15 pts** Missing or incorrect proof that a closed walk under 2*w(T) exists
- **- 20 pts** Completely incorrect, low-effort proof
- **- 15 pts** IDK

HW Solution

CS/ECE 374: Algorithms & Models of Computation, Spring 2019 Version: **1.0**

Submitted by:

- **≪Jiawei Tang≫**: **≪jiaweit2≫**
- **≪Pengxu Zheng≫**: **≪pzheng5≫**
- **≪Abhay Varmaraja≫**: **≪abhaymv2≫**

28

Solution:

28.A.

What we have here is a line-shape graph that every two vertices are connected with one edge. The ordering algorithm would be first we deploy the servers at the left-most and right-most vertices. Then we perform something like binary search where we deploy each server at the middle point. For example, we would deploy a server at the $\frac{n}{2}$ vertex for the first iteration. For the second, we would deploy servers at $\frac{n}{4}$ and $\frac{3n}{4}$ vertices, etc. We will have $log(n)$ iterations and each iteration will have cost $\frac{n}{2}$. Therefore, the total cost of deployment is $\Omega(nlogn)$.

28.B.

First, to prove that there exists a closed walk that visits all of vertices, we can use direct proof. Since now we know we have a minimum spanning tree T for the undirected graph G. Then we can be sure that there is a vertex v in G such that there is path from v to every other vertices. Because G is undirected, then there must exist a closed walk from v to every other vertices and when we are at the other vertices, we use the same edge to travel back to v. This closed walk will be consist of edges only in T if every other edges that are not in T have ∞ weight. Therefore, this closed walk will have $2w(T)$ weights and we can conclude that the total weight of edges of the walk is always at most $2w(T)$.

28.C.

Assume that there are at least two vertices in X. The closest pair of vertices, x, y, in X should be connected with each other with a single edge. Therefore, $d(x, y) = min(E)$ where E is the set of all edges in the graph. In a situation where all the vertices in X forms a cycle. Also because we have previously in part b, proved that there must be a closed walk visiting all vertices in X, we can say that the lower bound of weights of any closed walk in X is $d(x, y) \cdot |X|$. Also, we know from part b that the total weight of edges of a closed walk is at most $2w(T)$. Then we can conclude that

$$d(x, y) \cdot |X| \leq 2w(T)$$

28.D.(Discussed with Hengzhi's group)

By 28.C, we have proved that for the closest pair x,y of vertices in X, $d(x, y) \leq 2w(T)/|X|$. For each time the recursive greedy algorithm runs, there will be one less closest pair of vertices in X but the inequality above still holds. Then the total cost would be $\sum_{i=1}^{|X|} d(x_i, y_i)$. Because of the conclusion in 28.C, This total cost will be less than or equal to $\sum_{i=1}^{|X|} \frac{2w(T)}{|X|-i}$. We can see this sum is of harmonic series form as the $|X|-i$ can be replaced with i. By Jeff's notes(http://jeffe.cs.illinois.edu/teaching/algorithms/notes/99-recurrences.pdf) regarding harmonic series,

$$\sum_{i=1}^{|X|} \frac{2w(T)}{i} = 2w(T)H_{|X|} = \Theta(2w(T)log(|X|)) = \Theta(w(T)log(|X|))$$

Thus, the deployment cost is bounded by $O(w(T)log(|X|))$

3 28.C. **20 / 20**

✓ **- 0 pts** **Correct**

- 5 pts Minor error in proof

- 20 pts Incorrect proof

- 15 pts IDK

HW Solution

CS/ECE 374: Algorithms & Models of Computation, Spring 2019 Version: **1.0**

Submitted by:

- **≪Jiawei Tang≫**: **≪jiaweit2≫**
- **≪Pengxu Zheng≫**: **≪pzheng5≫**
- **≪Abhay Varmaraja≫**: **≪abhaymv2≫**

28

Solution:

28.A.

What we have here is a line-shape graph that every two vertices are connected with one edge. The ordering algorithm would be first we deploy the servers at the left-most and right-most vertices. Then we perform something like binary search where we deploy each server at the middle point. For example, we would deploy a server at the $\frac{n}{2}$ vertex for the first iteration. For the second, we would deploy servers at $\frac{n}{4}$ and $\frac{3n}{4}$ vertices, etc. We will have $log(n)$ iterations and each iteration will have cost $\frac{n}{2}$. Therefore, the total cost of deployment is $\Omega(nlogn)$.

28.B.

First, to prove that there exists a closed walk that visits all of vertices, we can use direct proof. Since now we know we have a minimum spanning tree T for the undirected graph G. Then we can be sure that there is a vertex v in G such that there is path from v to every other vertices. Because G is undirected, then there must exist a closed walk from v to every other vertices and when we are at the other vertices, we use the same edge to travel back to v. This closed walk will be consist of edges only in T if every other edges that are not in T have ∞ weight. Therefore, this closed walk will have $2w(T)$ weights and we can conclude that the total weight of edges of the walk is always at most $2w(T)$.

28.C.

Assume that there are at least two vertices in X. The closest pair of vertices, x, y, in X should be connected with each other with a single edge. Therefore, $d(x, y) = min(E)$ where E is the set of all edges in the graph. In a situation where all the vertices in X forms a cycle. Also because we have previously in part b, proved that there must be a closed walk visiting all vertices in X, we can say that the lower bound of weights of any closed walk in X is $d(x, y) \cdot |X|$. Also, we know from part b that the total weight of edges of a closed walk is at most $2w(T)$. Then we can conclude that

$$d(x, y) \cdot |X| \leq 2w(T)$$

28.D.(Discussed with Hengzhi's group)

By 28.C, we have proved that for the closest pair x,y of vertices in X, $d(x, y) \leq 2w(T)/|X|$. For each time the recursive greedy algorithm runs, there will be one less closest pair of vertices in X but the inequality above still holds. Then the total cost would be $\sum_{i=1}^{|X|} d(x_i, y_i)$. Because of the conclusion in 28.C, This total cost will be less than or equal to $\sum_{i=1}^{|X|} \frac{2w(T)}{|X|-i}$. We can see this sum is of harmonic series form as the $|X|-i$ can be replaced with i. By Jeff's notes(http://jeffe.cs.illinois.edu/teaching/algorithms/notes/99-recurrences.pdf) regarding harmonic series,

$$\sum_{i=1}^{|X|} \frac{2w(T)}{i} = 2w(T)H_{|X|} = \Theta(2w(T)log(|X|)) = \Theta(w(T)log(|X|))$$

Thus, the deployment cost is bounded by $O(w(T)log(|X|))$

4 28.D. 40 / 40

- ✓ **- 0 pts** Correct
- **- 10 pts** Minor error in proof
- **- 20 pts** Incorrect or unclear cost calculation to get O(w(T) log n)
- **- 40 pts** Completely incorrect, low-effort proof.
- **- 30 pts** IDK