

ADA Homework Three

Donald Hsu

December 2, 2023

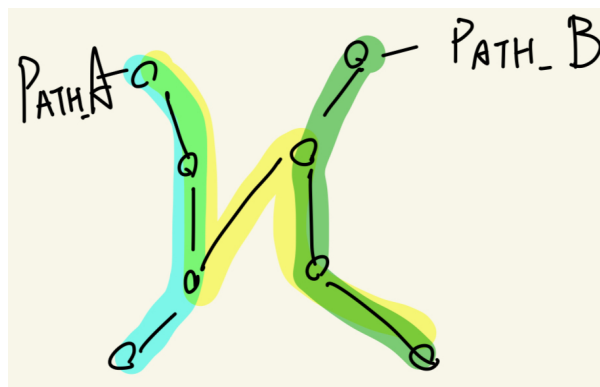
Problem 0

- **Problem 1:** b11902046 李明奕, b11902155 陳致和, b11902011 陳愷欣
- **Problem 2:** b11902084 張閔堯, b11902132 官毓韋, b11902158 黃昱凱
- **Problem 3:** b11902027 陳思瑋, b11902030 王瑜立
- **Problem 4:**

Problem 4

(a) Longest trip of Xian Chong Country

1. If there exist two longest PATH_A and PATH_B with length k in Xian Chong Country but they doesn't cross each other, but since that we can reach any island from any other island, there must be a way to link PATH_A and PATH_B on island i and j without passing any island already on PATH_A and PATH_B. By doing so, we can find that there should be a path that is longer than k that cross PATH_A and PATH_B as the picture below.



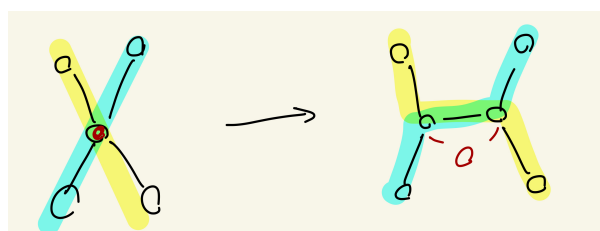
Therefore we know PATH_A and PATH_B must cross each other or pass through the same island.

Assume that Arctan follows PATH_A and Alexandar follows PATH_B, and Arctan passed island i in the n_{th} day while Alexandar passed island i in the m_{th} day and $n > m$.

We can find out that after passing the island, Arctan still has $k - n$ island to pass while Alexandar still has $k - m$ island to pass. But since $n > m$, $(k - n) < (k - m)$, Alexandar can get a path that is longer by follow the Arctan path after island i , and we can tell that when $n < m$ the assumption is still wrong, so we knew in this situation $n = m$.

and since that $n = m$ should be fulfilled if they want to go through the path in any direction, so we can say that $n = m = \frac{k}{2}$.

we can expand the island as PATH_O as the picture below.



by the picture above, we can knew that if they pass PATH_O in the same direction,

they will pass every island and bridge in the same day. If they want to pass `PATH_O` in the different direction, they will pass the bridge or island in the center of `PATH_O` at the same day.

Therefore, we knew that they will meet during the trip.

2. First we can do BFS from the root and find one of the longest path's end i and BFS from i to find another longest path's end j and the length of the longest path k (the bridges the path crossed).

Since that in the previous problem, we can tell that at center of every longest path they must pass the same node (bridge) in the center of them.

- If it is a node in the center, we can set every node to red, and start BFS from the center, when the node found out it's depth is $\frac{k}{2}$, the node know it is one of the longest path's end, and will change it's color to green. After a node changed it's color, it will tell it's parent node to change it's color to green.
- If it is a bridge in the center, we can set every node to red, we temporarily remove the bridge, and start BFS from the two ends of the bridge, when the node found out it's depth is $\frac{k-1}{2}$, the node know it is one of the longest path's end, and will change it's color to green. After a node changed it's color, it will tell it's parent node to change it's color to green.

After then, we can know if the node is on the longest path simply by checking if the color of the node is green.

correctness:

If there exist a node n that can't be found we can assume that the length from the one of the longest path end e (which got n between center and e) is shorter than $\frac{k}{2}$ or $\frac{k-1}{2}$ (if the center is bridge.), but it isn't possible since the middle point or the middle bridge had the same length to every end of the longest path.

time complexity:

In this algorithm, we did only did BFS($O(n)$) two time. Therefore the time complexity is $O(n)$.

3. First we can do BFS from the root and find one of the longest path's end i and BFS from i to find another the longest path's end j and the length of the longest path k (the bridges the path crossed).

Since that in the previous problem, we can tell that at center of every longest path they must pass the same node (bridge) in the center of them.

- If it is a node in the center, we can set every node to red, and start BFS from the center, and count the number n of the nodes that has the depth $\frac{k}{2}$. After then, we can get the numbers of longest path by calculating $n \cdot (n - 1)$
- If it is a bridge in the center, we can set every node to red, we temporarily remove the bridge, and start BFS from the two ends of the bridge, and count the number n_a, n_b of the nodes that has the depth $\frac{k-1}{2}$ from side a and side b of the bridge. After then, we can get the numbers of longest path by calculating $(n_a \cdot n_b) \cdot 2$.

correctness:

Since that every longest path will pass the center node/ bridge, if the center we only need to calculate the number of permutation of the end of the longest paths is a node the number of paths will be ${}_nP_2$, if the center is a bridge we know the end point at the bridge's same side can't form a longest path, therefore we only need to match the end of the longest path at the bridge's both side, after multiply $(n_a \cdot n_b)$ by two ($A \rightarrow B$ & $B \leftarrow A$) we can get the number of the longest path.

time complexity:

In this algorithm, we did only did BFS($O(n)$) two time. Therefore the time complexity is $O(n)$.

(b) Grid King and Albert

1.

2.