|  |  |  |
| --- | --- | --- |
| **Assignment Question number** | **Functions Used** | **Brief of approach taken** |
| **Question 1** |  |  |
| **Question 2** | Main function:  **improveDistance()** -> returns the blocked point requires for question 2 and the ratio of improvement  Function calls within main function:  **getPathforImproveDistance()** -> This function is similar to addBilateralRelationship() function. This returns an updated list of locations considering the bilateral paths too as per the last assumption for this question.  **findPaths()** -> returns output to getPath()  **computerPathDistance()** -> calculates the total distance of a path  **computeDistance()** -> calculates the distance between two adjacent points | Assumptions used:   * the optimum point to be blocked is the first point croc will travel to in the shortest path, since if that point is blocked the croc cannot go any further in that path. * New alternative path is the second shortest path and it is assumed the croc would travel on the second shortest path when the first shortest path is blocked * If there is only one possible path between two points then that path will be the shortest path and since there is no alternative path new alternative path distance will be 0 which would make the ratio of improvement an indefinite very large number (because any number divided by 0 is indefinite and very large) * If all paths have the same point as the next point travels from the source destination, blocking that point means blocking all possible paths. This would also make new alternative path distance 0 since there is no alternative path * If 16 is a neighbor of 15 but 15 is not a neighbor of 16. So, we are assuming this is a bilateral way. This means if we do not think in a bilateral way there is only 1 path from 15 to 18 (15, 16, 18). But since we assume it bilateral way there are more possible paths. |
| **Question 3** |  |  |