| Notes:(Record key insights from readings and discussions.) |
| --- |
| **week-12**  **Thursday:**  Questions:   1. What is the best case (fastest case) when searching via this algorithm? 2. The best case is that if we get the goal at middle of the given values. 3. What is the worst case (slowest case)? 4. The worst case is that if the given value is at first index or last index of the given values. 5. If the size of the data set doubles, what happens to the best and the worst case? 6. If the size of the data doubles then there will be no effect on the best case, but the worst case becomes more effort for finding the goal. 7. How would you have to change the binary Search method if the data set is not sorted? 8. If the given data set is not sorted in binary search method then we have to first sort the array and perform the operations because if the array is not sorted then the result may be not predicted. Sorting of any array can be achieved by “Arrays.sort(Arrayname);” method.   output for IterativeBinarySearch .java    output for IterativeSearch .java    output for RecursiveBinarySearch .java    reference code: <https://github.com/jawaharsd/IteratingData.git> |

| Deliverable Status | | | | |
| --- | --- | --- | --- | --- |
| Deliverables | What did you plan to accomplish | What did you actually accomplish | Size | Effort |
| Effort logger analysis | * Planned to conduct meeting with my teammate to know the progress of the work. * Planned to complete the project according to the tasks given in the class. * Planned to complete reading and copying of data before Tuesday. | * Conducted team meeting on the progress of the work. * Completing the project referring the tasks given in the class. * Will complete the reading and copying the data before Tuesday. | 30% | 1 hour |
|  |  |  |  |  |
|  |  |  |  |  |