**Guided Learning Plan for ArrayList**

**Class\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Name\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Lesson 5 ——Searching**

|  |  |  |
| --- | --- | --- |
| **Topic** | **details** | **Essential Knowledge** |
| Searching——  Sequential or Linear search | **Sequential or Linear search** typically starts at the first element in an array or ArrayList and looks through all the items one by one until it either finds the desired value and then it returns the index it found the value at or if it searches the entire array or list without finding the value it returns -1.  fig_6-4_pg338.png  Note:Sequential or linear search is the only method that can be used to find a value **in unsorted data**.  Quiz：  Which will cause the longest execution of a sequential search looking for a value in an array of integers?And which will cause the shortest execution？  A. The value is the first one in the array B. The value is in the middle of the array C. The value is the last one in the array D. The value isn't in the array  Programming    Exercise1：  Searching an ArrayList of Integer  **Hint：The same algorithms can be used with arrays or ArrayLists, but notice that size() and get(i) is used with ArrayLists instead of length and [i] which are used in arrays.**  Exercise2：searching an ArrayList of Double  What‘s the difference?  Exercise3：searching an ArrayList of String  What’s the difference? | |
| Searching——  Binary search | Note：Binary search can only be used if the data is **sorted**.  fig_6-4_pg341.png  Binary search keeps dividing the sorted search space into half. It compares a target value to the value in the middle of a range of indices. If the value isn’t found it looks again in either the left or right half of the current range. Each time through the loop it eliminates half the values in the search area until either the value is found or there is no more data to look at.   * Binary search calculates the middle index as left + right / 2 where left starts out at 0 and right starts out at the array length - 1 (the index of the last element). * Remember that integer division gives an integer result so 2.5 becomes 2. It compares the value at the middle index with the target value (the value you are searching for). * If the target value is less than the value at the middle it sets right to middle minus one. If the target value is greater than the value at the middle it sets left to middle plus one. * Otherwise the values match and it returns the middle index. * It also stops when left is greater than right which indicates that the value wasn’t found and it returns -1.   Exercise：  Complete the method to realize binary search  public int binarySearch(int[] elements, int target) {  } | |
| Runtimes | How do we choose between two algorithms that solve the same problem?  They usually have different characteristics and runtimes which measures how fast they run. For the searching problem, it depends on your data.  Binary search is much faster than linear search, especially on large data sets, but it can only be used on sorted data.   * Often with runtimes, computer scientist think about the **worst case behavior.** * With searching, the worst case is usually if you cannot find the item. * With linear search, you would have to go through the whole array before realizing that it is not there, but binary search is much faster even in this case because it eliminates half the data set in each step. * We can measure an informal runtime by just counting the number of steps.   Runtimes can be described with mathematical functions. For an array of size n, linear search runtime is a linear function, and binary search runtime is a function of log base 2 of n (or log n + 1 comparisons). This is called **the big-O runtime function** （**the big-O notation**）in computer science, for example O(log n) vs. O(n). You can compare the growth of functions like n and log2n as n, the data size, grows and see that binary search runs much faster for any n.  You should be able to calculate how many steps binary search takes for a given n by counting how many times you can divide it in half.  Quiz1  How many times would the loop in the binary search run for an array int[] arr = {2, 10, 23, 31, 55, 86} with binarySearch(arr,55)?  A. 2 B. 1 C. 3  Quiz2  If you had an ordered array of size 500, what is the maximum number of iterations required to find an element with binary search?  A. approximately 15 times B. approximately 9 times C. 500 times D. 2 times | |
| Summary  (ESSENTIAL KNOWLEDGE  ) | 1. There are standard algorithms for searching. 2. Sequential/linear search algorithms check each element in order until the desired value is found or all elements in the array or ArrayList have been checked. | |