|  |  |
| --- | --- |
| **Name: Marina Moshchenko** | **Date: TBA** |
| **Class: CS Programming** | **CS Standards:**  9-12.CT.4 Implement a program using a combination of student-defined and third-party functions to organize the computation.  The focus is on having students think about how to decompose a programming problem into functions and procedures, including working around the constraints imposed by specific functions or features provided in a library  9-12.CT.5 Modify a function or procedure in a program to perform its computation in a different way over the same inputs, while preserving the result of the overall program.  The focus is on understanding that the same abstract concept can be performed in different ways in a program, as long as the same inputs yield the same results  9-12.CT.6 Demonstrate how at least two classic algorithms work, and analyze the trade-offs related to two or more algorithms for completing the same task.  The focus of this standard is a high-level understanding that algorithms involve tradeoffs, especially related to memory use and speed. Students should understand that classic algorithms are solved problems that can be reused.  9-12.CT.7 Design or remix a program that utilizes a data structure to maintain changes to related pieces of data.  The focus is on updating the elements or components within a named instance of a data structure, without changing the value associated with the name itself.  9-12.CT.9 Systematically test and refine programs using a range of test cases, based on anticipating common errors and user behavior.  The emphasis is on perseverance and the ability to use different test cases on their programs and identify what issues are being tested in each case.  9-12.CT.10 Collaboratively design and develop a program or computational artifact for a specific audience and create documentation outlining implementation features to inform collaborators and users.  The focus is on the collaborative aspect of software development, as well as the importance of documenting the development process such that the reasons behind various development decisions can be understood by other software developers.  9-12.DL.2 Communicate and work collaboratively with others using digital tools to support individual learning and contribute to the learning of others.  Digital tools and methods should include both social and professional (those predominantly used in college and careers). Collaboration should occur in real time and asynchronously, and there should be opportunities for students to both seek and provide feedback on their thoughts and products. |
| **UNIT:**  **Algorithms** | **Period/Topic:**  Search Algorithms |

|  |
| --- |
| ***Focus (In the form of a question – How? or Why?):***  ***How do we implement the Binary Search Algorithm recursively into programming?***  ***How do we access the efficiency of search algorithms?*** |
| ***Learning Target(s) Explicitly Stated / Students will be able to (S.W.B.A.T.):***  Students will be able to implement Binary Search Algorithm recursively into programming language; access the efficiency of linear and binary search algorithms and compare them. |
| ***Academic Vocabulary:***  ***Search algorithms, linear search, binary search, iterative, recursive, run time, best case scenario, worst case scenario, average case scenario.*** |
| ***Start-Up Assessment For Learning (Determine Prior Knowledge): 5-7 min***  Consider the sorted array: {1,2,3,9, 11,13,17,25,57,90}. The target number is 57. Fill in the trace table to trace the values of variables and subarrays.    ***\*Class check time, one volunteer shares the results*** |

|  |  |
| --- | --- |
| ***Mini****-****Lesson + Guided Practice****:*  Recursive calls | ***Notes/Questions/Answers:*** |
| **Activity 1**. Use [this Link](https://www.youtube.com/watch?v=sr_bR1WwcLY) [1] to see the video animation of a binary search applied to a sequence of numbers with steps explained. Define the sequence of repetitive steps which can be applied to the array and each of the subarrays (halves) when binary search is applied.  Define a pseudocode of a binary search algorithm where the method can be invoked recursively.  ***\*Class check time, each group post their results to a shared platform (google classroom, slack, jamboard, etc)***  Students review other posts and compare them. |  |
| ***Mini****-****Lesson + Guided Practice****:*  Part 2. Live coding of binary search algorithm using recursive approach. Students are being prompted to participate and define each of the following line of code. | public static boolean binarySearchRecursive (int[] array, int target, int left, int right){  if (left > right){  System.out.println("The searched number is not in the array (binary recursive)");  return false;  }  int mid = (left + right) / 2;  if (array[mid] == target){ //base case  System.out.println("Recursive binary: the searched item " + target + " has an index: " + mid);  return true;  }else if (target < array[mid]){  return binarySearchRecursive(array, target, left, mid - 1);  } else{  return binarySearchRecursive(array, target, mid + 1, right);  }  } |
| **Activity 2**. Small groups of 3-4  Run the program for all test cases/inputs:   1. Just right 2. Too low 3. Too high 4. Not in the array)   Identify the following cases invoke the methods :   1. Best case scenario (base case-just right), shortest run time. 2. Worst case scenario – longest run time 3. Average case scenario.   ***\*Class check time, one group shares their results*** |  |
| **Activity 3.** Small groups of 3-4  Students use the given code to check the elapsed time of each method invoked on a particular input. Using the model code, students check the run time of Linear, Binary Iterative and Binary Recursive search algorithms to compare the run time. | public static void main(String[] args) {  int[] numArray = {1,2,3,4,5,6,7,8,9,1,0,11,12,13,14,15};  long start, elapsed;    start = System.currentTimeMillis();  int myTarget = binarySearchIterative(numArray, 3);  elapsed = System.currentTimeMillis()-start;  System.out.println("Elapsed time for binarySearchIterative: " + elapsed + " misiseconds");  System.out.println();  } |

|  |
| --- |
| ***Exceeding Standards Task:***  Scale the program to have different inputs for different test cases:  1-15, 1-300, 1-1000, 1-10,000, etc. Compare the run time. |
| ***At Standards Task:***  Complete all programming parts of the lesson to a properly functioning code. |
| ***Below Standards Task:***  Use the code from live coding portion of the lesson for detailed comments. Compare the run time of 3 methods. |

|  |  |
| --- | --- |
| ***Summary (What have you learned today?):***  Exit Ticket. 3-5 min  The graph below demonstrates linear search algorithm as a function O(n), blue line; and binary search algorithm as a function O(logn), red curve line. Using the graph explain:   1. In which cases can a linear Scan find the value faster than a binary Search? 2. In which cases is it the best to use a Binary Search? | Students are expected to attest to a scalability of the program, such as for small numbers (input) the linear search is more efficient, whether for large numbers (input) binary search is significantly more efficient than the linear one. |

|  |
| --- |
| ***Homework:*** |

Resources:

[1] <https://www.tutorialspoint.com/data_structures_algorithms/linear_search_algorithm.htm> Linear search

[2] <https://www.tutorialspoint.com/data_structures_algorithms/binary_search_algorithm.htm> Binary Search

[3] <https://www.khanacademy.org/computing/computer-science/algorithms/intro-to-algorithms/a/a-guessing-game> Khan Academy Number Guessing Game

[4] <https://www.youtube.com/watch?v=sr_bR1WwcLY> Binary search step-by-step animation (video)