**Lesson Planning Template**

**COSMOS EDUCATIONAL TOOLKIT:** Finding the shortest distance

|  |  |  |
| --- | --- | --- |
| **Grade/ Grade Band**: 6-7 | **Topic:** Dijkstra’s algorithm | **Lesson #** 1 **in a series of** \_\_1\_\_\_ **lessons** |
| **Brief Lesson Description**:  In computer science, problems that may sound completely different turn out to be similar when you think about how to solve them. On the surface, the game Pac-Man, your family tree, and driving from New York to San Francisco are not related, however, all of these problems involve route-finding or finding the shortest path. Generally, we think of the shortest path problem when talking about traveling. However, the shortest path problem doesn’t have to represent traveling anywhere at all. You could use a graph of a social network (your family tree) to figure out how closely connected you are to some other person. In wireless networks such as these that the COSMOS testbed are providing, finding the shortest path can save you a lot of time. This lesson has a strong connection to routing algorithms used in COSMOS testbed and on the Internet in general. Moreover, this lesson introduces students to how we use algorithms to find the best cost, efficiency, and running time.   * The COSMOS testbed uses routing protocols to send signals and connect devices. Dijkstra's Algorithm is used in the *Routing* protocols (including wireless communication), the mechanisms that allow information to be gathered and distributed. In Routing protocols, *routing* algorithms such as Dijkstra's Algorithm, is used to determine paths. In addition, students learn about r*outing* databases. Which is used to store information that the algorithm has discovered. Students can find the shortest way to send a message from one node to another using the COSMOS Testbed. Where 1 node will be the router. | | |
| **Specific Learning Outcomes:**   * discuss their pre-existing knowledge of algorithms * Understand the definition of algorithms * work in groups to discuss and create algorithms * define and implement algorithms  apply Dijkstra’s algorithm to a network  * Analyze an algorithm for efficiency  trace back through a network to find a route corresponding to a shortest path | | |
| **Narrative / Background Information** | | |
| **Prior Student Knowledge Required:**  Shortest Path Problem  Have you ever used a Maps to find the shortest route to your destination or least amount of time to get from one point to another? If so, then you're already familiar with the shortest path problem. In math terms, this is a way to find the shortest possible distance between two vertices (or nodes) on a graph.Suppose we're trying to find the shortest path from your house to Jessica’s house. You know the distances between different locations in your neighborhood. These different locations are called vertices and the routes (path) between them are called edges, we can create a weighted graph representing the situation. Dijkstra's algorithm is a step-by-step process we can use to find the shortest path between two vertices in a weighted graph. This algorithm enables us to find shortest distances and minimum costs, making it a valuable tool. In today's lesson we will explore the Shortest Path problem, by performing a hands on activity and then solving the problem with pencil and paper. Afterwards, students will be introduced to the famous Dijkstra’s Algorithm that solves the shortest path problem. This lesson has a strong connection to routing algorithms used on the Internet. This lesson also introduces ideas about how we analyze algorithms: looking for correctness, efficiency and running time. Dijkstra's algorithm is a step-by-step process we can use to find the shortest path between two vertices in a weighted graph. This algorithm enables us to find shortest distances and minimum costs, making it a valuable tool. | | |
| **Problem Solving Practices (Ex: Standards for Mathematical Practice):**  **Standards for Mathematical Practice » Make sense of problems and persevere in solving them.**  Mathematically proficient students start by explaining to themselves the meaning of a problem and looking for entry points to its solution. They analyze givens, constraints, relationships, and goals. They make conjectures about the form and meaning of the solution and plan a solution pathway rather than simply jumping into a solution attempt. They consider analogous problems, and try special cases and simpler forms of the original problem in order to gain insight into its solution. They monitor and evaluate their progress and change course if necessary. Older students might, depending on the context of the problem, transform algebraic expressions or change the viewing window on their graphing calculator to get the information they need. Mathematically proficient students can explain correspondences between equations, verbal descriptions, tables, and graphs or draw diagrams of important features and relationships, graph data, and search for regularity or trends. Younger students might rely on using concrete objects or pictures to help conceptualize and solve a problem. Mathematically proficient students check their answers to problems using a different method, and they continually ask themselves, “Does this make sense?” They can understand the approaches of others to solving complex problems and identify correspondences between different approaches. CSTA K-12 Computer Science Standards (2011) CT - Computational Thinking Computer Science Principles 4.1 - Algorithms are precise sequences of instructions for processes that can be executed by a computer and are implemented using programming languages. 4.2 - Algorithms can solve many but not all computational problems. | **Main Content Ideas:**   * Point out that there are different ways to move between vertices and other elements such as time or distance, which influence the popularity of each route. * Getting from point A to point B can be represented by a graph * Dijkstra's algorithm finds the shortest path from point A to point B and returns the distance * In computer science “solving a problem” usually doesn’t mean coming up with the answer to a specific problem. Solving a problem means​ describing the algorithm​­ the series of steps ­that will solve​ every​ instance of a particular​ type​ of problem, regardless of the specifics. This is one of the things that makes computer science is different from other subjects. In computer science, rather than following a procedure to solve a problem, you get to ​invent ​the procedure. | **Possible Multidisciplinary Concepts:**  Math:   * students can apply this knowledge in math class to find vertices on a graph. * Students can discuss the mathematical concepts relating to Dijkstra algorithm. * Social Studies/ Geography Dijkstra algorithm is used in geographical Maps. students can apply this knowledge when learning how to read a map.   Science:   * Generate and compare multiple possible solutions to a problem based on how well each is likely to meet the criteria and constraints of the problem. |
| **Possible Preconceptions/Misconceptions:**   1. In computer science “solving a problem” usually doesn’t mean coming up with the answer to a specific problem. Solving a problem means​ describing the algorithm​­ the series of steps ­that will solve​ every​ instance of a particular​ type​ of problem, regardless of the specifics. This is one of the things that makes computer science is different from other subjects. In computer science, rather than following a procedure to solve a problem, you get to ​invent ​the procedure. It’s the difference between knowing the answer to 1+1, and describing how addition, works in general. To do that, you must really understand all aspects of the problem, so that your procedure can handle many different situations. As long as all the situations have the same fundamental properties in common, your algorithm should be able to solve any of them. 2. The Dijkstra algorithm uses labels that are positive integers, which have the strict weak ordering defined (numbers are tied to each other). 3. Two nodes can be directly connected but it doesn't mean it's the fastest way to send information.   Example:   |  |  | | --- | --- | | On the Internet, of course, we’re interested in finding the fastest way to route information from one place to another. The graph below shows the connections between routers on a small segment of a network, and the amount of time (in milliseconds) it takes a packet to get between them.  If we are in charge of router A, we want to know the fastest way to get information from router A to router C. But, just because A is directly connected to C doesn’t mean it’s the fastest way to get information there. We need to find the shortest path.  The shortest path from A to C is highlighted. |  | | | |
| **LESSON PLAN – 5-E Model** | | |
| [**ENGAGE: Opening Activity – Access Prior Learning / Stimulate Interest / Generate Questions:**](http://www.youtube.com/watch?v=PUB1GU_tvpI&safe=active)  Activity #1 Pass the string  Play the game degrees of separation (connection) with yarn   * Start the game by sitting all the students in a circle with 2-3 students sitting in the center and 2-3 students sitting outside the circle. * Give one student, one end of yarn or string and ask them to pass it to anyone in the circle. * Repeat the second step until everyone is connected by the string * Ask the students, how are they connected to another student?   Activity #2 Plan your Journey  When you plan a journey, there are different factors you might consider:   * Do you want to go the shortest distance? * Do you want to take the minimum time? * Do you want to minimise the cost?   Activity #3 Find the shortest route  Create a map below on the floor using masking tape and safety cones. Ask students to find the shortest route. | | |
| **EXPLORE: Lesson Description – Materials Needed / Probing or Clarifying Questions:**   1. Yarn or sting (10-20 meters long) 2. Masking tape 3. Safety cone 4. color pencils 5. [Instructions on how does Dijkstra's Algorithm work](https://www.youtube.com/watch?v=eFZCPlZCyIM) 6. [What is an Algorithm worksheet](https://docs.google.com/document/d/1pG4HNObGxIwnFKmelZknJ-tS2QRNIZ8AlBVVCU9e4r0/edit)  [Dijkstra's Algorithm worksheet](https://docs.google.com/document/d/1WUZ24OfdMf7jIfgAyttO7e9brspqfmsvZlOXrBa4Dv8/edit)   1. [Vocabulary Worksheet](https://docs.google.com/document/d/1fNiTnOdPb0OQTgA_7AWOdg5LaSi2tozabLxCQdMDXXk/edit) | | |
| **EXPLAIN: Concepts Explained and Vocabulary Defined:**  **Key Vocabulary:** Algorithm: An algorithm is a set of instructions designed to perform a specific task.A Nodes is a device or data point in a larger networkShortest Path: the lowest weighted path between two nodes. weights that represent the cost, distance, and time data we want to consider.Efficiency: a measure of the number of steps per input size needed to complete an algorithma weighted graph is a collection of vertices and edges with edges having a numerical value (or weight) associated with them. | | |
| **ELABORATE: Applications and Extensions:** | | |
| **EVALUATE:**  **Formative Monitoring (Questioning / Discussion):**   1. Activity #1 Pass the string 2. What is an Algorithm worksheet 3. Activity #2 Plan your Journey 4. Activity #3 Find the shortest route   **Summative Assessment (Quiz / Project / Report):**   1. [Dijkstra's Algorithm worksheet](https://docs.google.com/document/d/1WUZ24OfdMf7jIfgAyttO7e9brspqfmsvZlOXrBa4Dv8/edit) 2. [Vocabulary worksheet](https://docs.google.com/document/d/1fNiTnOdPb0OQTgA_7AWOdg5LaSi2tozabLxCQdMDXXk/edit) | | |
| **Elaborate Further / Reflect: Enrichment:**   1. What were some of the most interesting discoveries I made while working on this project? About the problem? About myself? About others? 2. How well did I and my team communicate overall? 3. What were some things my teammates did that helped me to learn or overcome obstacles? 4. Briefly summarise the skills that you think you have developed or improved during the lesson. 5. How will I use what I've learned in the future? | | |