1.1.1) **Sorting Example:**

Organizing a music festival schedule by sorting bands based on their popularity ensures the most popular bands play at peak times, maximizing audience engagement.

Shortest Distance Example:

Planning a delivery route for a community outreach program involves finding the shortest path between schools to minimize travel time and distance, similar to solving the Traveling Salesman Problem.

1.1.2) **In a real-world setting, other than speed, you might need to consider:**

1. Resource Utilization: How efficiently an algorithm uses memory, CPU, or other resources.

2. Scalability: How well the algorithm performs as the size of the input grows.

3. Energy Consumption: Especially important for mobile and battery-powered devices.

4. Reliability: Consistency in producing correct results.

5. Maintainability: How easy it is to understand, modify, and maintain the algorithm.

6. Cost: Financial cost associated with running the algorithm, especially in cloud computing environments.

1.1.3) **Binary search trees (BSTs).**

Strengths:

1. Efficient Searching: BSTs allow for efficient searching, insertion, and deletion operations, typically with a time complexity of O(log n) in a balanced tree.

2. Ordered Data: They maintain elements in a sorted order, which is useful for in-order traversal to get elements in ascending order.

3. Versatility: They can be used to implement other data structures like sets and maps.

Limitations:

1. Unbalanced Trees: If the tree becomes unbalanced (e.g., all elements are inserted in ascending order), the time complexity can degrade to O(n) for searching, insertion, and deletion.

2. Complexity: Implementing self-balancing BSTs (like AVL trees or Red-Black trees) can be complex and require additional memory and time for maintaining balance.

3. Space Overhead: Each node requires additional memory for storing pointers to its children, which can be significant in memory-constrained environments.

1.1.4) **Similarities:**

1. *Graph-Based Problems*: Both the shortest-path and traveling-salesperson problems are discussed within the context of graphs, where nodes represent locations and edges represent paths.

2. *Optimization Goals*: Both problems aim to optimize a certain criterion. The shortest-path problem seeks the shortest route between two points, while the traveling-salesperson problem seeks the shortest possible route that visits each point exactly once and returns to the starting point.

Differences:

1. *Problem Scope*: The essay mentions that the shortest-path problem deals with finding the shortest route between two specific points. On the other hand, the traveling-salesperson problem requires finding a tour that visits all points and returns to the starting point.

2. *Complexity and Solution*: The shortest-path problem has known efficient algorithms (like Dijkstra's algorithm) to solve it. In contrast, the traveling-salesperson problem is NP-complete, meaning no efficient algorithm is known to solve it for all instances, and it often requires approximation algorithms.

3. *Nature of the Solution*: The shortest-path problem results in a single path between two points, while the traveling-salesperson problem results in a cyclic tour that visits all nodes.

So, we can see that while both problems involve finding optimal paths in graphs, the traveling-salesperson problem is much more complex due to its NP-complete nature and requirement to visit all nodes in a cyclic manner. NP(nondeterministic polynomial time)

1.1.5) **Problem Requiring the Best Solution:**

- Medical Diagnosis and Treatment Planning: In critical healthcare situations, such as diagnosing a life-threatening disease like cancer, only the best and most accurate solution will do. Doctors need the most precise diagnosis to determine the exact type and stage of cancer to plan the most effective treatment. Any approximation could lead to incorrect treatment, potentially endangering the patient's life.

Problem Where an Approximate Solution is Good Enough:

- Route Planning for Delivery Trucks: For a delivery company planning routes for trucks, an approximate solution that minimizes the overall distance traveled is often good enough. While the exact shortest possible route (traveling-salesperson problem) might be ideal, using an efficient approximation algorithm can save significant time and computational resources, providing a near-optimal solution that is practically sufficient for daily operations.

1.1.6) **Let's consider the scenario of weather forecasting and disaster response:**

Entire Input Available:

- When planning for seasonal weather patterns, meteorologists often have complete historical data and long-term climate models available in advance. This allows them to make predictions and prepare for expected conditions like average temperatures, rainfall patterns, and seasonal storms.

Input Arrives Over Time:

- During a natural disaster, such as a hurricane or a severe storm, new data continuously arrives from satellites, weather stations, and sensors. Emergency response teams must make real-time decisions based on this incoming data to issue warnings, evacuate areas, and deploy resources, without knowing all future developments in advance.

This example illustrates how sometimes all necessary information is available upfront, while other times decisions must be made dynamically as new data arrives.