Solutions for Introduction to Algorithms 2022, 4th edition.
link to MIT open course

1.1.1 Describe your own real-world example that requires sorting. Describe one that requires finding the shortest distance between two points.

Sorting product tables by various parameters: date, price, etc. AABB collisions needs to know shortest distance between two points in rectangle-circle case.

1.1.2 Other than speed, what other measures of efficiency might you need to consider in a real-world setting?

Memory.

1.1.3 Select a data structure that you have seen, and discuss its strengths and limitations.

Hash table. Operations `Search`, `Delete` and `Insert` take $\mathcal{O}(1)$ time on average. However, database can degrade if it goes through a large number of collisions.

1.1.4 How are the shortest-path and traveling-salesperson problems given above similar? How are they different?

Both problems regard path finding with minimum cost. In traveling-salesperson we considers multiple graph vertices, while in shortest-path we deal with two vertices.

1.1.5 Suggest a real-world problem in which only the best solution will do. Then comeup with one in which "approximately" the best solution is good enough.

We would want best solution for airport scheduling. Approximately the best solution will fit any route optimization problem.

1.1.6 Describe a real-world problem in which sometimes the entire input is available before you need to solve the problem, but other times the input is not entirely available in advance and arrives over time.

Any problem with complete dataset is an example where the entire input is avaliable. One problem where input is not entirely avaliable would be stock market price prediction.

1.2.1 Give an example of an application that requires algorithmic content at the application level, and discuss the function of the algorithms involved.

Graphical algorithms for GUI in web and mobile applications.

1.2.2 Suppose that for inputs of size n on a particular computer, insertion sort runs in $8n^2$ steps and merge sort runs in $64n\lg n$ steps. For which values of n does insertion sort beat merge sort?

$$8n^2 \leq 64n\lg n$$
$$n \leq 8\lg n$$
$$n - 8\lg n \leq 0$$
$$2^n - n^8 \leq 0$$
$$n \in [1.1; 43.6]$$

1.2.3 What is the smallest value of n such that an algorithm whose running time is $100n^2$ runs faster than an algorithm whose running time is 2^n on the same machine?

$$100n^2 < 2^n$$
$$n < 14.3$$

Problems