Activity 1: Algorithms

Each week, you will work in teams of 3–4 students to learn new concepts. This activity will introduce you to the process. We'll use a simple game to explore basic searching algorithms.

Model 1 Hi-Lo Game

Hi-Lo is a number guessing game with simple rules, played by school children.

a) There are two players – *A* and *B*.
b) Player *A* thinks of a number from 1 to 100.
c) Player *B* guesses a number.
d) Player *A* responds with "too high", "too low", or "you win".
e) Players *B* and *A* continue to guess and respond until *B* wins (or gives up).

Questions (20 min)

Start time: _____

1. How many different answers can player *A* give?

Three (too high, too low, you win).

2. When does the game end?

The game ends when *B* guesses the correct number or gives up.

3. Play the game a few times to ensure that everyone understands the rules.

4. Identify 4–5 different guessing strategies that Player *B* could use. Each strategy should describe a **different approach** to the game. For example: *Start at 1, and count up until the correct answer is found.* In computer science, we call such strategies **algorithms**. Try to have a mixture of simple and clever algorithms, including ones that young children could use.

a) Randomly guess numbers until you get the right one.

b) Start at 100 and count down by 1 to the correct answer.

c) Count up by 10 until too high, and then count down by 1.

d) Count up by 25, then count down by 5, and finally count up by 1.

e) Start at 50, if too high search 1–49, or if too low search 51-100, etc.

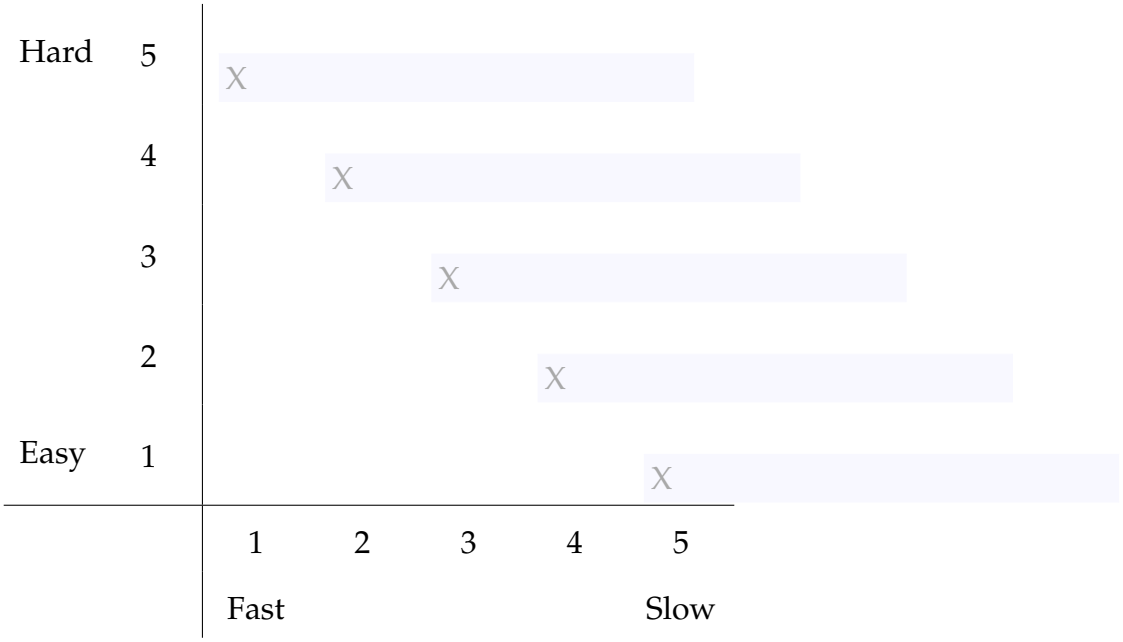
5. Rank order the algorithms with regard to how **fast** they will find the right answer. Write 1 for the fastest algorithm (fewest guesses) and 5 for the slowest one (most guesses).

In this particular example:
1: e 2: d 3: c 4: b 5: a

6. Rank order the algorithms with regard to how **easy** they are to describe or specify. (Suppose you had to explain them to a first-grader so that he/she could play the game.) Write 1 for the algorithm that is easiest to describe and 5 for the one that is hardest.

In this particular example:
1: a 2: b 3: c 4: d 5: e

7. For each algorithm (*a* to *e*), plot its fast and easy values on the graph:



8. In complete sentences, describe the relationship between the fast and easy rankings, including what you see from the graph.

Algorithms that are really easy to describe (and are not that sophisticated) generally run the slowest. On the other hand, fast algorithms are complex and more difficult to describe.

In computing, we often must search for a particular item in a set. Computer scientists are particularly interested in searching very large sets, with thousands or millions of values. For example, the Harvard University Library has roughly 16,000,000 volumes, and the US Library of Congress has roughly 22 million cataloged books and over 100,000,000 total items.