Algorithms - Drawing Challenge

With your 3 or 4 students sitting near you, create instructions that another team could follow to draw a simple object. Do the following:

1. Introduce Yourself (5 minutes) - Introduce yourself to your team, where you are from, and what you like doing outside of your academics.

2. Decide on a simple object to draw (10 minutes) - It should be something that needs only about 8-12 strokes of a pencil. Try to choose something challenging to test your instruction-giving abilities.

3. Write Instructions (15 minutes, finish by 2:35pm) - On a piece of paper write a clear and precise set of instructions that another team could follow to draw this object. Your instructions may only use words. You cannot use pictures. You cannot use words for any shapes to describe what to draw.

4. At 2:35pm, Follow Another Team's Instructions (10 minutes) - Trade your instructions with another team near you. See if you can follow that team's instructions to draw their object. Draw what you think the object is on the instructions page, and note the problems encountered.

Drawing Challenge Reflection

- What specifically made the instructions poor or ineffective?

- What could you do next time to increase the likelihood that someone could better follow your instructions and succeed in drawing the object?

What is an algorithm?

What are the key characteristics of a “good” algorithm?

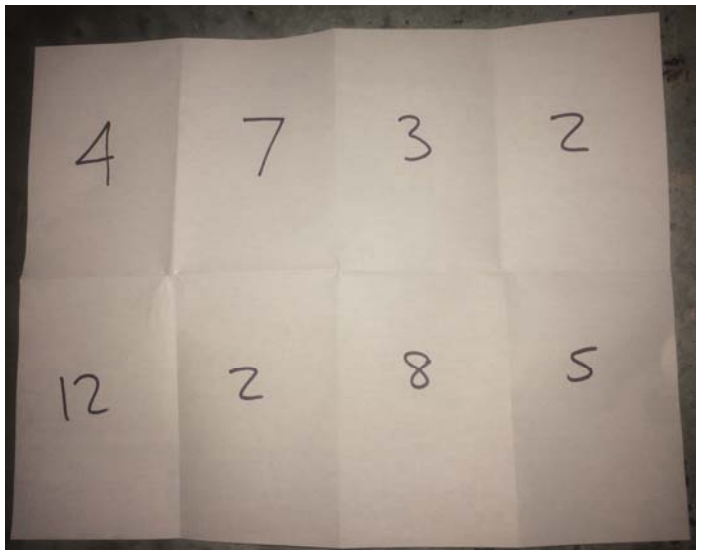
- Algorithms are well-ordered.
- Algorithms have unambiguous operations.
- Algorithms have effectively computable operations (right level of detail).
- Algorithms produce a result (hopefully correct).
- Algorithms halt in a finite amount of time.
- Algorithms must “scale”

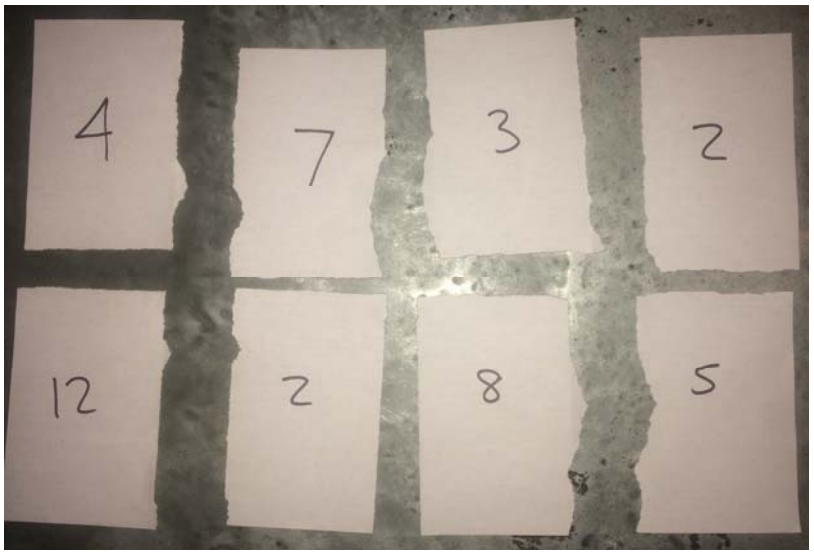
Algorithmic Approaches

- Exhaustive – try all possibilities, not usually the best approach, but helpful to understand the size of the solution space.
- Intelligent – There are various algorithm approaches that are “more efficient” than an exhaustive approach. You may already be aware of these intelligent algorithmic approaches, and use them in your daily life, without realizing it. The ability to use a particular intelligent algorithmic approach depends on the problem. Some examples of intelligent algorithmic approaches that you will learn about in later classes are greedy, divide and conquer, and dynamic programming.

Lab 02 Preview

- Bring 8 playing cards to lab, or make your own 8 pieces of paper with random numbers from 1-15 on them.





Complete the CS 100
Lecture 02 Survey
- Entry Survey

Login to IIT Gmail
Midnight Deadline