CPSC 121: Models of Computation

Unit 0
Introduction

George Tsiknis
Based on slides by Patrice Belleville and Steve Wolfman

Introductions

- Instructor: George Tsiknis

  Office: ICICS 307
  Email: tsiknis@cs.ubc.ca
  Office Hours: on the course web site

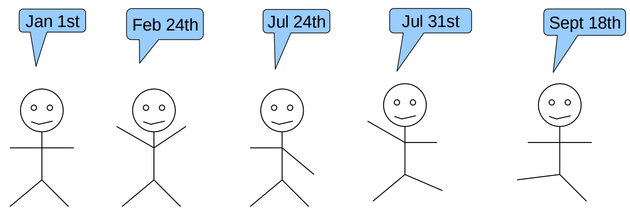
- TAs : See course site.

Learning Goals

- By the end of this unit, you should be able to:
  - Give an example of how we can apply formal reasoning to a simple, real-world task.
  - Give an example of how a computational solution to this simple task might go wrong.
  - Describe the four “big questions” which we will address in CPSC 121.

Activity

- Find an algorithm to order students by birthday.



Problem

- How many swaps did you need to make?

Question

What is the maximum number of swaps we may make to order 4 people by their birthday?

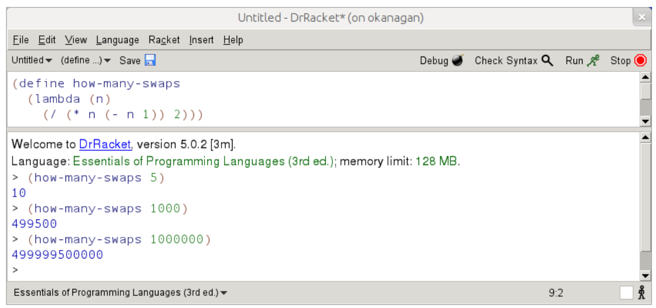
a. 3
b. 4
c. 6
d. 12
e. None of these

How many swaps?

- For 2 people?
- For 3 people?
- For 4 people?
- For 5 people?
- …
- For `n` people?

Computing Swaps with DrRacket

- Computing n(n-1)/2 using DrRacket:



Computing Swaps in Java

- Computing n(n-1)/2 using Java:

```java
import java.io.*;

public class Compute
{
    public static void main(String[] args)
    {
        int n = Integer.parseInt(args[0]);
        System.out.println(n * (n-1) / 2);
    }
}
```

```
poirot> java Compute 5
10

poirot> java Compute 1000
499500

poirot> java Compute 1000000
-364189984
```


Questions Answered in CPSC 121:

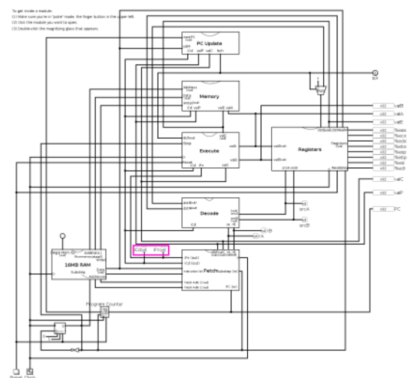
- How can we prove that n(n-1)/2 is the largest number of swaps needed for n birthdays?
  - Can use the method of *Mathematical Induction*
- Why did our Java implementation print a negative value, but not the Racket implementation?
  - Use different *Number representation*

? CPSC 121: The BIG questions: ?

1. How can we convince ourselves that an algorithm does what it's supposed to do?
2. How do we determine whether or not one algorithm is better than another one?
3. How does the computer (e.g. Dr. Racket) decide if the characters of your program represent a name, a number, or something else? How does it figure out if you have mismatched " " or ( )?
4. How can we build a computer that is able to execute a user-defined program?

Our Working Computer

- A working computer you will learn about in the labs:



Course Learning Outcomes

- After you complete the course, you will be able to:
  - model important problems so that it is easier to discuss, reason about, solve, and test them.
  - learn new modeling formalisms more easily.
  - communicate clearly and unambiguously with other CS experts on complex topics.
  - characterize algorithms (CS problem solutions), by proving their correctness or efficiency.
  - critically read proofs: justifying why each step is correct and judging what the proof means.
  - explain how computers work.

Course Administration

Updated on Jan 4

- Explore the CPSC 121 website: http://www.ugrad.cs.ubc.ca/~cs121/current
  - You are ***required*** to be familiar with the course website.
  - Check the announcements daily
- Read carefully the ***Course Information*** document on the course web site.
- Check the Connect site for the course:
  - Pre-class Quizzes
  - Marks
- Check the Piazza site for the course discussion board

Pre-Class Learning Goals for Next Lecture

By the start of next class, you should be able to:

- Translate back and forth between simple natural language statements and propositional logic.
- Evaluate the truth of propositional logic statements using truth tables.
- Translate back and forth between propositional logic statements and circuits that assess the truth of those statements.

First Quiz

- The first online quiz is due ______________________.
- Sections to read for the quiz:
  - Epp, 4th edition: 2.1 and 2.4.
  - Epp, 3rd edition: 1.1 and 1.4
  - Rosen, 6th edition: 1.1 up to the top of page 6, and 11.3.