Section 1: Introduction

This first section introduces us to thinking about problems *algorithmically*. For the time being, we are just solving problems the "hard way" by doing things manually, or trial-and-error. Later on, we will look at more mathematical approaches for solving these.

One of the purposes of studying math – even if you don't see yourself using, say, *calculus* in your software programs in the future – is to train problem solving skills. Part of computer science and software development is about building solutions to problems.

4 Steps to Problem Solving

1. Understanding the problem

- Can you state the problem in your own words?
- What are you trying to find or do?
- What are the unknowns?
- What information do you obtain from the problem?
- What information, if any, is missing or not needed?
- Don't impose conditions that do not exist

2. Devising a plan

- Is this problem similar to another problem you have solved?
- Can one of the Problem Solving Strategies be used?

3. Carrying out the plan

- Implement the strategy or strategies in step 2, and perform necessary actions or computations.
- Check each step of the plan as you proceed. This may be intuitive checking or a formal proof of each step.
- Keep an accurate record of your work.

4. Looking back

- Check the results in the original problem. (In some cases this will require a proof.)
- Interpret the solution in terms of the original problem. Does your answer make sense? Is it reasonable?
- Determine whether there is another method of finding the solution.
- If possible, determine other related or more general problems for which the techniques will work.
- Make a point of thinking about the strategy that finally worked for this type of problem for future reference.

Section 2: The Josephus Game

In computer science and mathematics, the Josephus problem (or Josephus permutation) is a theoretical problem related to a certain counting-out game.

People are standing in a circle waiting to be executed. **Counting begins at a specified point** in the circle and **proceeds around the circle in a specified direction**. **After a specified number of people are skipped, the next person is executed**. The procedure is repeated with the remaining people, starting with the next person, going in the same direction and skipping the same number of people, until only one person remains, and is freed.

The problem — given the number of people, starting point, direction, and number to be skipped — is to choose the position in the initial circle to avoid execution.

(From https://en.wikipedia.org/wiki/Josephus_problem)

Example

We begin with a circle of 10 people, numbered 1 to 10. Every-other-person is executed, starting with person 2 *(obviously you don't want to be person #2.)* Where is the last person standing?

Beginning

Step 1

Step 2

Step 3

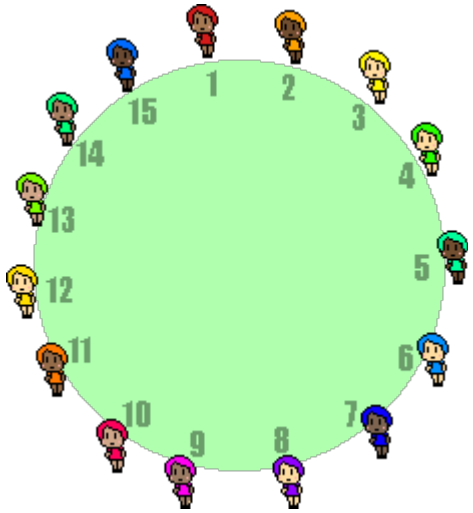
Step 4

Step 5

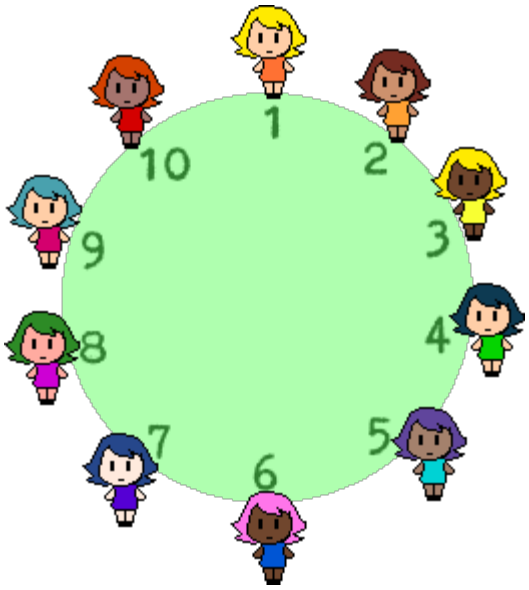
And we can keep going on. Who is the last man standing?

Exercise 1 **50%**

(a) Given a Josephus circle of 15 people, if we are eliminating every 3rd person (starting with person 3), who is the last to be tagged – and the 2nd to last to be tagged?



(b) Given a Josephus circle of 10 people, if we are eliminating every 4th person (starting with person 4), who is the last to be tagged – and the 2nd to last to be tagged?



Section 3: Game Trees

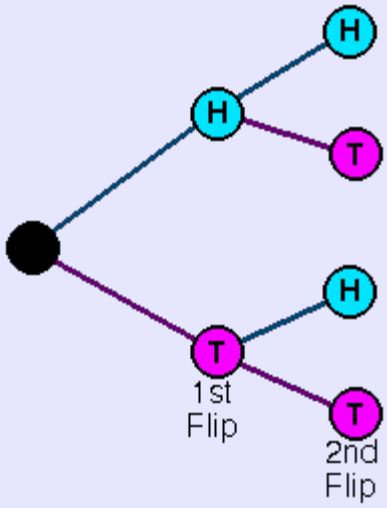
In section 1.1, we will also be looking at events that have multiple outcomes – such as flipping one coin, two coins, or three coins, or who of two people win one, two, or three tennis matches. With small amounts of "variables", we can list out all the possible outcomes, and we can build a game tree based on this.

Example

If you flip one coin, the result will be HEADS (H) or TAILS (T).

If you flip two coins, what are all the outcomes?

- ***HH***
- ***HT***
- ***TH***
- ***TT***



Exercise 2 **50%**

Suppose you toss three coins – a nickel, a dime, and a quarter, and you record the results in that order. For example, "HTH" means nickel = heads, dime = tails, quarter = heads.

(a) In a systemic way, list all the different results you could record.

(b) Draw a game tree for recording the results.

(c) On the game tree, label each possible result either 0, 1, 2, or 3, indicating how many *heads* it has.

(d) Do you think a person who tosses three coins is more likely to get all three heads, or to get exactly two heads?