

# Parallel Programs are Fast!



## **Objective:**

To compare different computer programs for sorting a list of numbers and decide which is fastest. You will play the role of the computer, programmed to solve this problem.

## **Introductory Remarks:**

Sorting is a simple, but fundamental problem. Imagine being given all the names of the people in Columbus and having to put them in alphabetical order. A computer program can do this by comparing pairs of names and switching them if they are out of order. There are many different programs for sorting.

A sequential program does one thing at a time. It's like having one chef in the kitchen. A parallel program does lots of things at the same time. It's like having a whole team of chefs in the kitchen. The meal will be prepared more quickly, but the chefs need to be organized and coordinate with each other.

## **Materials:**

- A stack of 50 “number” cards (8 ½ x 11) each with a random number between 1 and 500.
- A set of 15 “position” cards (8 ½ x 11) number 0 to 14.
  - Even/odd cards are different colors (0, 2, 4... are red, 1,3,5,... are blue).
  - Position cards are set out in a single row of positions, in increasing order.
- 15 participants, each stands next to a “position” card and holds a “number” card.

## **Activity:**

### *Sequential Program: “Bubble Sort”*

1. Compare numbers at positions 0 & 1. If number cards are out of order, participants are swapped.
2. Repeat for positions 1 & 2, then 2 & 3, then 3 & 4, until the end of the row.
3. Which number is at the end of the row?
4. Repeat this several times and you should see the list becoming sorted.

### *Parallel Program 1: “Even-Odd Transposition Sort”*

1. Even phase: Participants on red position face right, participants on blue position face left. Participants switch places if their numbers are out of order.
2. Odd phase: Participants on red positions face left, participants on blue positions face right. Participants swap places if their numbers are out of order.
3. Repeat both even & odd phases until list is sorted.

### *Parallel Program 2: “Radix Sort”*

1. Each number is held up in turn, so that everyone can see it.
2. Each participant keeps count of how many numbers are less than her own.

3. After the last number has been shown to everyone, each person walks directly to the place card that matches her count. The list is sorted!

**Challenges:**

1. Which sorting program was fastest? Why?
2. Write your own parallel sorting program. How fast would it be?