Objectives:

Learn binary numbers and counting (how computers count).
Learn what comparative sorting algorithms are with examples.

Plan:

1) http://csunplugged.org/sites/default/files/activity_pdfs_full/unplugged-01-binary_numbers.pdf

Work through the binary counting worksheets for 20 minutes (until 4:20). I think we should be able to get up to the “secret message” page, at least.

2) http://csunplugged.org/sites/default/files/activity_pdfs_full/unplugged-07-sorting_algorithms.pdf

Work through the sorting algorithms worksheets for 20 minutes (until 4:40).

Use a shuffled stack of number cards instead of weights. For example, for the quicksort example, shuffle the cards, then draw the first one to use as your pivot, then place the rest of the cards into stacks on either side per the algorithm.

If we can have at least two students per group, then have the students split up tasks - one does the comparison between two cards, e.g. deciding that a new 3 card is less than a pivot of 5. The other is in charge of putting it in the right location in the new list. This makes it a bit more like a computer - the students only look at a few things at a time, rather than trying to sort the list all at once their own way.

I think we should be able to get up through trying insertion sort and bubble sort. Make sure to have the students count how many comparisons are necessary for selection sort versus quicksort.

Sorting algorithm cheat sheet.
For our purposes, a ‘list’ is just cards laid flat next to one another.

selection sort: find the smallest number in the stack (comparing two at a time), add it to the right of the new list. Repeat until out of cards.

quicksort: pick a random pivot - e.g. the top card on the stack. Make two new stacks on either side - numbers smaller than the pivot to the left, numbers larger to the right. Repeat this process for each stack until all the cards are laid out individually.

insertion sort: sort as you go - take the top card and make it the start of the new list. Insert all subsequent cards into the right positions as you go - e.g. start at the left and find when the new card is greater than the card on the left and less than the card on the right in the new list, then insert it there.

bubble sort: lay all of the shuffled cards out in a list. Starting at left, compare each neighboring cards. If they are out of order (left is bigger than right), switch them. Repeat at each position until you've gone through the list. Repeat this entire procedure (start at the left, and progress to the right) until no switches have to be made.

3) Final activity A: timing selection sort versus quicksort. Each group does one of each type of sorting while an adviser times them (using a smartphone). We then put up the times on the whiteboard and take the average.

4) (optional) FInal Activity B: Least significant digit radix sort. This combines what they learned about binary numbers with a sorting algorithm. The algorithm is: sort all the numbers by their least significant digit, then by the next most significant digit. Keep in mind that for this algorithm, a number like 2 has infinite 0s before it, for sorting purposes. (e.g. 02 is sorted to the left of 12 if sorting by the tens place).

Example (underlines are positions that move at the next step, bars represent new groups):

[1, 24, 31, 75, 86, 95, 101]
[1, 31, 101, | 24, | 75, 95, | 86] (grouped at 1s place)
[1, | 24, | 31, | 75, | 86, | 95, | 101] (grouped at 10s place)
[1, 24, 31, 75, 86, 95, | 101] (grouped at 100s place - no change)