Sorting Recap and Analysis

CSSE 221

Fundamentals of Software Development Honors

Rose-Hulman Institute of Technology

ROSE-HULMAN INSTITUTE OF TECHNOLOGY

Announcements

- Cycle 2 due Monday.
- I hope to check your Cycle 3 user stories today.
- Exam on Thursday is optional
  - Programming only, worth 50 points
  - Gives more time to focus on project

Searching & sorting are ubiquitous

- In the classic book series *The Art of Computer Programming*, Donald Knuth devotes a whole volume (about 700 pages) to sorting and searching.
  - Claims that about 70% of all CPU time is spent on these activities.

- You need sorting to do fast search

Elementary Sorting Methods

- **Selection sort**
- **Insertion sort**
- **Merge sort**
- Quicksort
- Binary tree sort
- Heapsort
- Radix sort
- And **lots** of others (see Wikipedia)

Goals:

1. How does each work?
2. Best, worst, average time?
3. Extra space requirements?

1. Selection Sort

- ***Idea:*** Select smallest, then second smallest, …
- What's the runtime?
  - Best?
  - Worst?
  - Average?
- Extra space?

```
n = a.length;
for (i = 0; i < n; i++) {
    minPos = 0;
    // find the smallest
    for (j = i; j < n; j++){
      if (a[j] < a[minPos]){
            minPos = j;
    }
    // move it to the end
    swap(a, i, minPos);
}
```

Interlude: A 5-year old's understanding of swapping

- Courtesy of my son Caleb…

2. Insertion Sort

- ***Idea:*** Like sorting files in manila folders

- What is the runtime?
  - Best?
  - Worst?
  - Average?
- Extra space?

```
n = a.length;
for(i = 1; i < n; i++){
    temp = a[i];
    j = i;
    while (j>0 && temp<a[j-1]){
        a[j] = a[j-1];
        j--;
    }
    a[j] = temp;
}
```

3. Merge Sort

- ***Idea:*** Recursively split the array, then merge sorted sublists

- What is the runtime?
  - Best?
  - Worst?
  - Average?
- Extra space?

```
n = data.size();
if (n <= 1) { return data; }
int middle = n / 2;
left = data.subList(0, middle));
right = data.subList(middle, n);
// recursively sort each half
left = mergeSort(left);
right = mergeSort(right);
// merge sorted lists
return merge (left, right);
```

4. Quicksort

- Recursive, like mergesort
- If length is 0 or 1, then it's already sorted
- Otherwise:
  - Pick a "pivot"
  - Shuffle the items around so all those less than the pivot are to its left and greater are to its right
  - Recursively sort the two "partitions"

Interesting questions…

- Arrays.sort:
  - If objects, merge (since *stable*)
  - If primitives, quick (since faster)
  - Cuts over to insertion sort when n <= 7
- What would a recursive selection sort look like?
- How can we re-use sorting methods when we want to sort by different keys?

Project time

- In a few minutes...

Videos for upcoming C Unit

- We start C on Monday
- We will use an *inverted classroom* to help your productivity
  - What's that mean?
  - One downside for this weekend…
- You are free to pair-program the assignments
- You can bring headphones to class

Project time

- Show me what you've done recently:
  - Status report on cycle 2 user stories
  - Demo your program to me
- Show me what you are working on next
  - Cycle 3 user stories