7 October

Searching

- **binary search**: locate target value in sorted array by recursively eliminating half the array
  - expected runtime?
    - Notice: element at index N/2 can be found in 1 comparision
    - element at N/4, 3N/4 can be found in 2 comparisons
    - need x comparisons to find all of 2^(x-1) elements
    - if we have k operations to find N=2^(k-1) elements, k=O(log(N))
  - another way:
    - after k operations, we will have N/2^k elements remaining in array
    - task is solved when we have one element left: N/2^k <= 1
    - number of operations is proportional to log(N)
- built-in method to Java `Array` class, but array must be sorted first
  - returns index, or negative (index where the element *would* have been + 1)
- pseudocode of looping algorithm

```
public static int binarySearch(int[] a, int target) {
  int min = 0;
  int max = a.length - 1;

  while (min <= max) {
    int mid = (min + max) / 2;
    if (a[mid] < target) {
      min = mid + 1;
    } else if {a[mid] > target) {
      max = mid -1;
    } else {
      return mid;
    }
  }

  return -(min+1);
}
```

- psuedocode of recursive algorithm

```
public static int binarySearch(int[] a, int target) {
  return binarySearch(a, target, 0, a.length - 1);
}

private static int binarySearch(int[] a, int target, int min, int max) {
  if (min > max) {
    return -(min+1);
  } else {
    int mid = (min + max) / 2;
    if (a[mid] < target) { // too small, go right
      return binarySearch(a, target, mid+1, max);
    } else if (a[mid] > target) { // too large, go left
      return binarySearch(a, target, min, mid-1);
    } else {
      return mid;
    }
  }
}
```

Sorting

- fundamental in computer science / software engineering
- many solutions
- to sort, need comparison functions: $<$, $>$, `compareTo`

Example: Selection Sort

- look through list to find smallest value. Swap to index 0.
- look through to find second-smallest. swap to index 1.
- ...
- Repeat until all values are in proper places.

Runtime?

$$(n-1) + (n-2) + ... + 1 = \sum_{i=1}^{n-1} i = (n-1)\frac{(n-1)+1}{2} \tag{1}$$

9 October

Sorting

Insertion Sort

- very similar to selection sort
- iterate over array, building sorted array at the beginning
- compare unsorted values to largest value in sorted array
    - will be adjacent as list/array is built
    - starting condition: assume first element is sorted
- compare a[i] to next element a[i+1]
    - if a[i+1] < a[i], **insert** a[i+1] at the correct location in sorted list
    - if a[i+1]>a[i], move to next element a[i+2] (done if we find end of array)

Worst case run-time? Reverse sorted array

$$(n-1) + (n-2) + ... + 1 = \sum_{i=1}^{n-1} i = (n-1)\frac{(n-1)+1}{2} \tag{2}$$

- Does better on "nearly sorted" arrays: If each element is at most k places away from its sorted position, we get O(kn)
- This is called an *adaptive sorting algorithm*: performance increases with "sortedness" of array
- Other sorting algorithm properties:
    - *stability*: leaving order of equal elements unchanged
    - *in place*: only requires O(1) additional memory
    - *online*: can sort a list as it recieves it

Takeaway: - Insertion sort scans *backward* from current key - Selection sort scans *forward* - Both build sorted array "as they go," selection sort is guaranteed to sort the k smallest elements after k passes, while insertion sort sorts input in-order

Which one would you choose to adapt for an *online* sorting scenario?

Bubble Sort

- first described on paper in 1956!
- Donald Knuth says we shouldn't teach you this!

psuedocode for array `a`:

```
n = a.length
swapped = true

while swapped:
    swapped = false
    for(int i = 1; i < n; i++):
        if a[i-1] > a[i]: // wrong order
            swap(a[i-1], a[i])
            swapped = True

return a
```

- worst case: O(n**2). example: reverse sorted array as input
- not used in practice, except in computational geometry / graphics for situation where you have *nearly* sorted arrays
- highly parallelizable

Visualizer for Selection, Insertion, Bubble Sort

- https://visualgo.net/en/sorting
- plus Obama video

Homework questions?