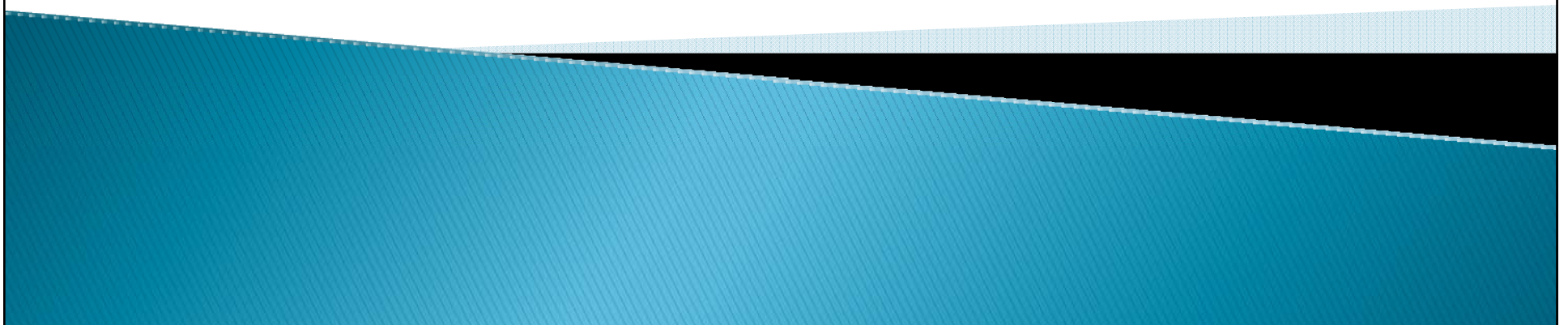


CSSE 220 Day 25

Exam Retrospective

Sorting intro

Work on Spellchecker Project



CSSE 220 Day 25

- ▶ If HW25 written done early, can submit now, or under my door before 8:05 tomorrow

- ▶ Questions?
- ▶ Today:
 - Exam retrospective
 - Sorting intro
 - Meet your Spellcheck partners and get organized; start your design

Exam results

Programming Part

Score	Number of students
51+	12
45-50	1
40-44	1
35-39	1
30-34	2
25-29	2
<25	9

Written Part

Score	Number of students
45-50	4
40-44	9
35-39	4
30-34	4
25-29	4
<25	1

Total

Score	Number of students
90-100	12
80-89	1
70-79	2
60-69	5
50-59	2
<49	6

The grades were bi-modal, so I will likely lower the percentage required for the lower grades in the gradebook's grading scale.

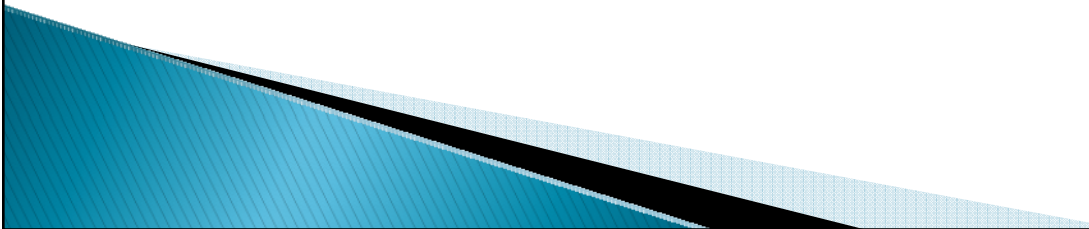
Course grades at this point

Grade	Number of students
A	7
B+	3
B	6
C+	2
C	5
D+	0
D	4
F	1

Written exam problems

- ▶ I will go through each problem quickly.
 - If you need more explanation of any of them, please ask questions.
 - Some of these are likely to reappear on final exam
- ▶ Then quick review of programming problems
 - Another data structs one is likely to appear on exam.

Sorting Intro

- ▶ What do we mean by "sort"?
 - ▶ What is the best sorting algorithm?
 - ▶ The three very simple Algorithms
 - Bubble Sort
 - Why is it so slow?
 - Insertion sort
 - Selection sort
 - ▶ Inversions and movement
 - ▶ Faster algorithms
- 

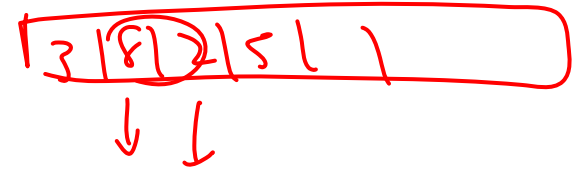
Elementary Sorting Methods

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

Goals:

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

Intro: Swapping



- Recall that calling `swap(a[i], a[j])` on

```
swap(int x, int y) {  
    int temp = x;  
    x = y;  
    y = temp;  
}
```

Call by value

x=2 y=8

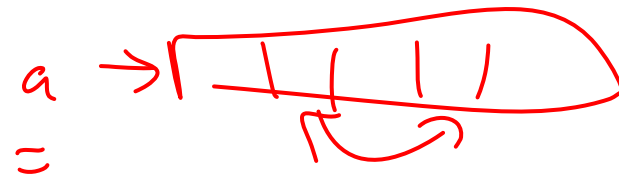


doesn't work! (Why?)

- Instead call `swap(a, i, j)` on

```
swap(int[] a, int i, int j) {  
    int temp = a[i];  
    a[i] = a[j];  
    a[j] = temp;  
}
```

~~no new int [6]~~



Extra space?

$O(1)$

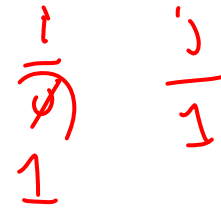
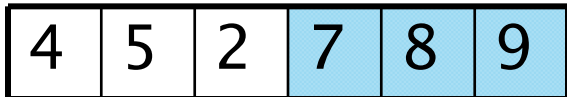
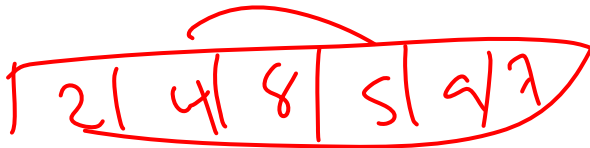
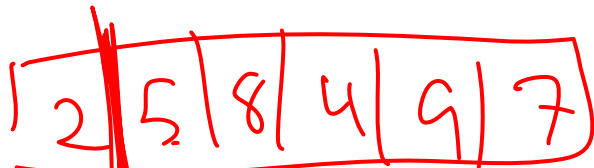
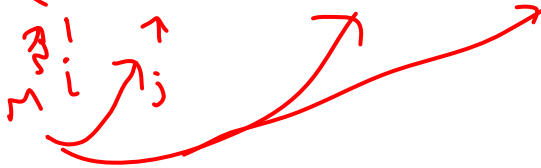
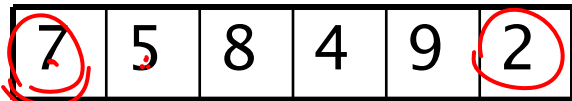
1. Selection Sort

- ▶ *Idea:* Select smallest, then second smallest, ...

<http://www.cs.oswego.edu/~mohammad/classes/csc241/samples/sort/Sort2-E.html>

```
→ n = a.length
  for (i = 0; i < n-1; i++) {
    → minPos = 0
      // find the smallest
      for (j=i+1; j < n; j++){
        if (a[j]<a[minPos]){
          minPos = j
        }
      }
      // move it to the start
      swap(a, i, minPos)
    }
```

1. Selection Sort



```
n = a.length
for (i = 0; i < n-1; i++) {
    minPos = 0
    // find the smallest
    for (j=i+1; j < n; j++){
        if (a[j]<a[minPos]){
            minPos = j
        }
    }
    // move it to the start
    swap(a, i, minPos)
}
```

After outer loop repeats 3 times:

9 comparisons, but only 3 swaps (9 assignments)

1. Selection Sort

► What's the runtime?

- Best? $\Theta(n^2)$
- Worst? $\Theta(n^2)$
- Average? $\Theta(n^2)$

► Extra space?

- 5 = n-1
- 4 = n-2
- 3 = n-3
- ⋮
- 1 = n-(n-1)

$$\frac{(n-1)(n)}{2} = \Theta(n^2)$$

$$\Theta(1)$$

```
n = a.length
for (i = 0; i < n-1; i++) {
    minPos = 0
    // find the smallest
    for (j=i+1; j < n; j++){
        if (a[j]<a[minPos]){
            minPos = j
        }
    }
    // move it to the start
    swap(a, i, minPos)
}
```



SpellChecker and Suggester

- ▶ Other projects have been highly-specified. For this one, you have a lot of leeway and can be very creative.
- ▶ GUI-based program
- ▶ Check the words of a text file for spelling
 - User can browse to file
- ▶ Flag words that are not in program's dictionary
- ▶ Suggest possible alternate spellings
 - Think of ways misspelling can occur:
 - missing or added letters
 - transposed letters
 - no space between words
 - things you come up with
- ▶ An interface that allows user to correct the spelling.
 - change, ignore, ignore all, ...

SpellChecker and Suggester

- ▶ Some GUI things you'll want to learn how to do
 - Browse to a file and open it
 - Deal with text in a text box
 - Display a list of choices and get user selection
- ▶ Some things to do if you didn't do them already.
 - Look for a dictionary to use (share it!)
 - Look at user interfaces of some spell-checkers
 - Look up various Java classes that may be useful
 - Especially helpful: The Java Swing book from Safari Tech Books online (see course syllabus)

Mini-project timetable

- ▶ Now. Look for a dictionary, think about the kinds of spelling errors you want to detect/correct.
- ▶ Day 25. Begin working with your partners.
- ▶ Day 27. Demonstrate some progress in class.
- ▶ Day 30. Final submission of the project is due.

Spell-checker teams

Repository

csse220-200830-spell001 Members:kleinjt,wanstrijm,harriska

csse220-200830-spell002 Members:drososmj,decluecm,schulte

csse220-200830-spell003 Members:baekj,strayeta,stoverre

csse220-200830-spell004 Members:stanlead,swansom1,carlsojs

csse220-200830-spell005 Members:brousapg,wisejl,hollanbm

csse220-200830-spell006 Members:cobbba,huntdz,segolp

csse220-200830-spell007 Members:chaddhd,cranemd,kotsybja,warnerbc

csse220-200830-spell008 Members:watersbt,robertic,chelmirs

csse220-200830-spell009 Members:borcheb1,ksiazecg,mathisjp

