About Kirkwood

• Community college serving 7-county region surrounding Cedar Rapids, Iowa
• Over 20,000 annual credit class enrollment
• Third or fourth largest college in Iowa
• Open enrollment policy
About CS @ KCC

– Credit program in Math/Science Department, part of Arts/Sciences Division

– One full-time instructor/coordinator (me)

– Service population includes CS majors (about 120) and Engineering majors (about 250)
CS courses

– CS1: 4-6 sections per year, 20-25 students per section

– “Upper level” courses – small, mostly CS majors:
  • Data Structures
  • Software Design & Development
  • Computer Organization & Assembly Language
  • Discrete Math
Assess why?

– Teaching is my job and I know what I’m teaching – but how do I know what they’re learning?
– Assessment is part of professional accountability
Assess why?

– Articulation: my students start with me, but they finish elsewhere

– Assessment provides evidence to help convince potentially skeptical audience (transfer institutions) that my students are prepared for their programs
Assess why?

– Assessment is part of institutional continuous improvement process

– Internal: 5-year C&I departmental reviews

– External: NCA process
Assess what?

– Specific learning outcomes
– Overall performance in terms of:
  • course
  • program
  • gen ed goals
Assess how?

– Tie exam questions to specific objectives
– Example objective from Data Structures: syllabus
  • The student will give examples of situations in which one sort [algorithm] or another is a more or less suitable solution to the problem at hand.
2. (15 points) **Rank the following combinations of searches and sorts**, given sorted or unsorted data. You may **assume that the order of magnitude** (big O) for the entire operation (sort/search combination) is the sum of the big O values for the algorithms involved. For ranking, use 1 to indicate the fastest combination, and 5 for the slowest. **Show the big O value you used for each sort or search algorithm in the box listing each algorithm.** Be sure to take into account whether or not the data set is presorted. **Use the average case for each algorithm if initial sort state doesn’t make any difference.**

<table>
<thead>
<tr>
<th>Initial sort condition</th>
<th>Sorting algorithm</th>
<th>Search algorithm</th>
<th>Rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unsorted</td>
<td>Selection sort</td>
<td>Serial search</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Big O:</td>
<td>Big O:</td>
<td></td>
</tr>
<tr>
<td>Unsorted</td>
<td>Heap sort</td>
<td>Binary search</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Big O:</td>
<td>Big O:</td>
<td></td>
</tr>
<tr>
<td>Sorted</td>
<td>Insertion sort</td>
<td>Serial search</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Big O:</td>
<td>Big O:</td>
<td></td>
</tr>
<tr>
<td>Sorted</td>
<td>Quick sort</td>
<td>Binary search</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Big O:</td>
<td>Big O:</td>
<td></td>
</tr>
<tr>
<td>Unsorted</td>
<td>Merge sort</td>
<td>Serial search</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Big O:</td>
<td>Big O:</td>
<td></td>
</tr>
</tbody>
</table>
Assess how?

— Detailed rubrics for grading assignments
— Excerpt from CS1 program rubric:

**Primary traits**
In order to receive a perfect score, a program must:
• be free of errors
• meet or exceed functional specifications and produce required output
• adhere to documentation and style guidelines given in class

**PTA**
Error-free (35% of score)
5 -- program runs consistently and is free of run-time errors when executed with appropriate test data
4 -- program compiles and runs without error for most test data; may fail or produce inaccurate results under certain conditions
3 -- program does not compile because of minor syntax/structural errors
2 -- program compiles but fails consistently with run-time errors
1 -- program is incomplete and/or exhibits serious errors of structure and syntax
Assess how?

– Clickers: instant assessment tool
– Students answer questions embedded in lecture notes
  • immediate feedback, trouble check
  • keeps students engaged
Results tracking
– Graph below shows enrollment in CS1 against success (A/B) vs. failure (F/drop) since clicker adoption (Fall 2006):

![Graph showing enrollment in CS1 against success (A/B) vs. failure (F/drop) since clicker adoption (Fall 2006).]
Why **not** assess?

- Time & labor intensive
- Results may not be definitive, clearcut, or to my liking
- Virtue has to be its own reward – no real incentives for doing this
- Conclusion: for me, it’s worth it