COP 5536 Advanced Data Structures

Summer 2012

Instructor: Dr. Sartaj K Sahni (sahni@cise.ufl.edu)

Office: E536 Computer Science and Engineering Building

TA: Min Chen (min@cise.ufl.edu)

Office: E309 Computer Science and Engineering Building

Office Hours: 3:00 – 5:00 PM Wednesday, Friday

Class Schedule:

Week 1(May 14 - May 18): Lecture 1 - Lecture 4

Week 2 (May 21 - May 25): Lecture 5 - Lecture 8

Week 3 (May 28 - Jun 1): Lecture 9 – Lecture 11   May 28 Memorial Day Holiday (no classes)

Week 4 (June 4 - June 8): Lecture 12 - Lecture 15

Week 5 (June 11 - June 15): Lecture 16 - Lecture 19

Week 6 (June 18 - June 22): Lecture 20 - Lecture 23

Week 7 (June 25 - June 29): Summer Break (no classes)

Week 8 (July 2 - July 6): Lecture 24 - Lecture 27

Week 9 (July 9 - July 13): Lecture 28 - Lecture 31

Week 10 (July 16 - July 20): Lecture 32- Lecture 35

Week 11 (July 23 - July 27): Lecture 36 - Lecture 39

Week 12 (July 30 – Aug 3): Lecture 40

Pre-requisites:

You should know the following:

1. C, C++, or Java. Since the text is in C++, you should at least be able to read C++.

2. Algorithm analysis methods (in particular asymptotic complexity).

3. Basic data structures such as stacks, queues, linked lists, trees, and graphs. Basic sorting methods such as insertion sort, heap sort, merge sort, and quick sort.

Course Requirements:

There will be three assignments and three exams. The exams will be closed book exams. The programming assignment(s) may be done in any high level language. Some possibilities are C, C++, and Java. Please have the use of any other language approved by the instructor or the TA. C++ is the preferred language.

Grading:

Exam 1: 25% (Thursday, Jun 14, 2012)

Exam 2: 25% (Thursday, July 12, 2012)

Exam 3: 25% (Monday, August 6, 2012)

Three Assignments: 25%

Course Outline

The specific topics and associated readings are:

1. Amortized complexity (Web)
2. External sorting & tournament trees (Sections 7.10.1, 7.10.2, and 5.8)
3. Buffering (Section 7.10.3)
4. Run generation & optimal merge patterns (Huffman trees) (Sections 7.10.4 and 7.10.5)
5. Priority queues and merging (Section 5.6)
6. Leftist trees, Binomial heaps and Fibonacci heaps (Sections 9.2, 9.3, and 9.4)
7. Pairing heaps (Section 9.5)
8. Double ended priority queues (Sections 9.6 and 9.7, Web)
9. Static and dynamic weighted binary search trees (Section 10.1)
10. AVL-trees (Section 10.2)
11. Red-black trees (Section 10.3)
12. Splay trees (Section 10.4)
13. B-, B+- and B*-trees (Sections 11.1-11.3)
14. Tries and digital search trees (Sections 12.1-12.3)
15. Tries and packet forwarding (Section 12.5)
16. Suffix trees (Section 12.4)
17. Bloom filters (Section 8.4)
18. Segment trees (readings)
19. Interval trees
20. Priority search trees (readings)
21. k-d trees (readings)
22. Quad and oct trees (readings)
23. BSP trees
24. R-trees

Lecture	Content	Reading
1	Amortized complexity.	Web resource.
2	Amortized Complexity.	Web resource.
3	Introduction to external sorting.	Section 7.10.1.
4	Introduction to external sorting.	Section 7.10.1.
5	Selection trees & k-way merging.	Sections 5.8 and 7.10.2.
6	Run generation.	Section 7.10.4.
7	Optimal merging of runs.	Section 7.10.5.
8	Buffering.	Sections 7.10.3.
9	Double-ended priority queues. General methods.	Sections 9.6, 9.7, and Web resource.
10	Double-ended priority queues. Interval heaps.	Sections 9.7.
11	Leftist trees.	Section 9.2.
12	Binomial heaps.	Section 9.3.
13	Binomial heaps.	Section 9.3.
14	Fibonacci heaps.	Section 9.4.
15	Pairing heaps.	Section 9.5.
16	Dictionaries.	Section 5.7.
17	Optimal binary search trees.	Section 10.1.
18	AVL trees.	Section 10.2.
19	AVL trees	Section 10.2.
20	Red-black trees.	Section 10.3.

21	Red-black trees.	Section 10.3.
22	B-Trees.	Sections 11.1 and 11.2
23	B-trees.	Sections 11.1 and 11.2.
24	B+ and B*-trees.	Section 11.3.
25	Splay Trees.	Section 10.4.
26	Splay Trees.	Section 10.4.
27	Binary Tries.	Section 12.1.
28	Compressed Binary Tries.	Section 12.2.
29	High-order Tries.	Sections 12.3 and Web Resource.
30	Tries and Packet Forwarding.	Section 12.5.
31	Suffix Trees.	Section 12.4.
32	Bloom Filters.	Section 8.4.
33	Segment Trees.	
34	Interval Trees.	
35	Priority Search Trees.	References.
36	Priority Search Trees.	References.
37	Multidimensional Search Trees.	References.
38	Quad Trees.	References.
39	BSP Trees.	
40	R-trees.	

Course Policies:

1. Every student is expected to follow the University of Florida Honor Code. (See http://www.dso.ufl.edu/STG/default.html)

2. Handouts, assignments, solutions, and others will be posted on Sakai. Students should check Sakai regularly, at least once per week.

3. When submitting homework for grading, your answers should be written neatly and contain an explanation that is clear and reasonably concise.

4. For distance students, you will have a 7 day window following the in-class exam date to schedule and complete exam with their proctor.