CS2010 – Data Structures and Algorithms II

Lecture 13 – Review

chongket@comp.nus.edu.sg

NUS
National University of Singapore
School of Computing

Consultation Slots

- Dr Chong Ket Fah
  - Starting recess week till exam for CS2010 (6th Dec)
  - Every Tuesday to Thursday you can drop by my office from 2pm to 5pm
  - For other times, you can email me to make appointment
- Any other teaching staffs of CS2010
  - They are all undergrad students and have exams too
  - Thus, please redirect questions to me ☺

Outline of the Last Lecture

1. Written Quiz 2 debrief
2. Final Assessment
3. Advertisement
4. Past Year Exam Questions

WRITTEN QUIZ 2 DEBRIEF

THE FINAL ASSESSMENT

Written Quiz 1+2 Remarks

Please avoid repeating the same mistakes, e.g.

- Blank answer in questions with obvious/trivial partial answers
- Getting stuck in one hard question for too long
- Long winded answers in early parts, less time in harder parts
- Not grounded in basic concepts yet…
- No need to give code/pseudo code for algorithms already taught in class/tutorial/lab, just mention them
- Give pseudo-code only for user defined algorithms/modifications to taught algorithms

Final Assessment (1)

- Open book, but here is my suggestion…
- Prepare a “data structure”:
  - Several sheets of A4 paper, as many as you need…
    - But not too many :O
  - Either handwritten or printed
  - Use whatever font size that is convenient for you
- Purpose: O(**N**) → O(**1**)

Example “data structures”

Data Structures	Operations	Runtime	Key points of DS	…
Binary Heap	1. ShiftUp 2. ShiftDown 3. Insert 4. ExtractMax/Min 5. Fast Heap create …	1. O(lgN) 2. O(lgN) 3. O(lgN) 4. O(lgN) 5. O(N) …	1. Implement PQ 2. Basically a binary tree of height O(lgN) 3. Uses compact array to store items …	…
…	…	…		…

Algorithms	Problems solved	ADT used	Keys points of algo	Optimization	…
Prim’s O(ElogV)	1. MST on a weighted connected graph 2. Minimax problem	Priority Queue	1. Start from source vertex 2. Dequeue vertex with smallest edge from PQ add to MST 3. Enqueue neighbors with edge weight if neighbor not already in MST 4. Repeat 2 & 3 until PQ empty	Stop when all vertices included in MST …	…
…	…	…	…	…	…

Final Assessment (2)

- Topics: From the entire semester **with focus** on things from Lecture 5 onwards.
- Section 1 – The basics
  - Fill in the blank questions (mainly use your A4 sheets here + a little thinking), they are *basics* and used as time differentiator
- Section 2 – Analysis
  - Analysis questions like in WQ1&2. Also used as time differentiator
- Section 3 – Harder application questions
  - 4 questions with subtasks
  - All questions have mixture of easy (usually the earlier subtasks) and hard parts (usually the later subtasks)
  - **TIP: There will be 1 DP Question ….**

Final Assessment (3)

- Time Management
  - You have 120 minutes
  - **Time may be tight, thus you have to pace yourself well.**
  - I *assume* that you have O(**1**) searching performance with help of your few pages of A4 sheets for Section 1 ☺
    - This kind of O(**1**) searching performance is needed to clear the earlier "time differentiator" questions as fast as possible…
    - Do not spend more than **15 minutes** in Section 1!
  - I believe you all will have time to read all problems
    - Note: Do not skip the sub-questions in each question, some are easier to give you marks or they are meant to guide you
  - Don't spend too much time on hardest subtask for Section 3 (those marked with * in the square bracket containing the marks)
    - Give your best answer in an appropriate amount of time then re-visit them when you have time leftover after doing the rest

Final Assessment (4)

- My suggestion on pacing
  - Section 1 (Basic) – 15 mins
  - Section 2 (Analysis) – 15 mins
  - Section 3
    - Question 1 – 15 mins
    - Question 2 – 25 mins
    - Question 3 – 25 mins
    - Question 4 – 20 mins
  - 5 mins left over to check your answers ….

Review of final scripts

- 11th December 3pm to 6pm at my office (COM2-02-66)
- You have the chance to review your scripts before they are submitted to the department

ADVERTISEMENT

Relevant Algorithm Modules After CS2010

- **CS3230: Design and Analysis of Algorithms**
  - (core module for CS students. Need to pass CS2010)
  - Less new algorithms + DSes
  - More rigorous grounding in proof of correctness and time/space complexity calculations
- **CS3233: Competitive Programming**
  - (Need at least A- for CS2010 and like algorithms + programming a lot!)

PAST YEAR EXAM QUESTIONS