CS 381 (LE1) Midterm Preparation

Instructor: Simina Brânzei

Steps to prepare for the midterm exam

- Read the slides (see resources on Piazza) and follow the schedule of topics done in class with the assigned readings, which can be found on Piazza under resources in the file "LE1_schedule.html".
- Summary of relevant reading material: [CLRS stands for the Cormen-Leiserson-Rivest-Stein book and KT stands for the Kleinberg-Tardos book]
- After you read a problem or topic, close the books/computer/phone/notes and test your understanding by (1) writing down a proof summary in your own words (e.g. in at most 1/3 or 1/2 a page) with the high level steps of the proofs), and (2) write down a complete proof in your own words. If you get stuck at any of these steps, then try to reconstruct the missing part on your own. If you cannot do so, then revisit the proof paying attention to the missing step; this means it was a subtle point of the proof that needs more attention.
- Study with your study group and test your understanding, e.g. by quizzing each other on explaining these concepts or proofs, writing proofs at the whiteboard/blackboard explaining to your study group the proof as if you were presenting it to them for the first time, and solving together additional exercises from the books.

Readings

- All the slides from class, except the challenge problems. Reading the topics we learned from KT/CLRS is also recommended. Suggested chapters below:
- KT chapter 2, especially 2.1 and 2.2 to understand what it means for an algorithm to run in polynomial time.
- CLRS chapter 3 (Growth of functions)
- CLRS chapter 4 (Divide-and-conquer), in particular from the beginning of 4 to the end of 4.1, as well as 4.3, 4.4, 4.5, and 4.6 (up to the end of 4.6.1).
- If you need a recap of random variables and expectation, see CLRS chapter 5.
- CLRS chapter 7 (quicksort)
- CLRS chapter 9.2 (randomized selection)
- CLRS chapter 9.3 (median-of-medians)

- KT chapter 5, including 5.1 (mergesort), 5.3 (counting inversions), 5.4 (closest pair of points).
- KT chapter 7 (network flow). Max-flow Min-cut theorem, Ford-Fulkerson, applications.

Exam instructions

- The exam will take place on Thursday (October 6) in Hiler Thtr from 8 to 9:30 PM EST. Plan to bring your Purdue ID card. Exception: If you submitted a DRC request, the exam will take place separately and the time and location will be announced in a separate note.
- This is a closed book exam. No electronic devices, books, notes, or other written materials may be brought to the exam.
- There will be a lecture on Tuesday at the usual time, which will focus on solving additional exercises and reviewing for the exam.

Practice problems

Several suggested exercises to help you prepare for the exam are given next.

Problem 1. *Asymptotic notation, growth of functions, induction:*

- *KT chapter 2: solved exercises 1-8.*
- *CLRS chapter 3: exercises 3.1-1, 3.1-2, 3.1-3, 3.1-4, 3.1-6, 3.2-1, 3.2-8. Problems: 3-1, 3-2, 3-3, 3-4, 3-5.*

Problem 2 (Stable matching)**.** *State the stable marriage problem and the Gale-Shapley algorithm. Show that the algorithm runs in polynomial time (what is the worst case number of iterations?), that it produces a matching where each person is matched with exactly one other person. Show the resulting matching is stable. Run the algorithm on a 4X4 instance of your choice, showing the intermediate steps.*

Divide and Conquer

Problem 3. *Recall the quicksort algorithm from class.*

1. *Describe the quicksort algorithm where the pivot is chosen uniformly at random.*
2. *Argue that the algorithm sorts correctly on each input.*
3. *Write a 1/2 page proof summary showing the main steps for how to derive an upper bound on the expected number of comparisons.*
4. *Write a complete proof showing an upper bound of $O(n \log n)$ for the expected number of comparisons.*
5. *What are the best and worst case number of comparisons made by quicksort?*

6. *Answer the same questions (1-6 above) for quickselect, except the bound on the number of comparisons will be $O(n)$.*

Problem 4.
- *KT: Work through the solved exercises in KT chapter 5 (starting with page 242). Also exercises 1, 2, 3 from KT at the end of chapter 5.*
- *CLRS: chapter 4, exercises 4.5-1, 4.5-3, 4.5-4 and problems 4-1, 4-3, 4-5.*

Greedy algorithms

Problem 5. *Kleinberg-Tardos, Greedy algorithms chapter (4):*

- *Solved exercises 1,2, and 3.*
- *Exercises 1, 2, 8, 21, 29.*

Network flow and matchings

Problem 6.
- *Define the following concepts: flow, value of a flow, residual network, augmenting path.*
- *State the Ford-Fulkerson algorithm from class. Execute it on a 5 node instance of your choice, showing the intermediate steps and residual graph at each stage. What can we infer about the set of nodes reachable from the source at the end of the algorithm?*
- *What is the worst case runtime of the algorithm?*
- *State the max-flow min-cut theorem and write a 1/3 proof summary, explaining the main steps of the proof.*
- *State and prove the flow value lemma.*
- *State and prove weak duality for flows.*
- *Show how to use the Ford-Fulkerson algorithm to obtain a bipartite matching in a graph. What is the runtime of the algorithm on such an instance?*
- *Show how to solve the Hackaton problem (from class, see slides on flow applications) by using network flow.*
- *Give an algorithm to find the maximum number of edge disjoint $s - t$ paths in a network.*

Problem 7. *Kleinberg-Tardos:*

- *Solved exercises 1, 2, and 3.*
- *Exercises 1, 2, 3, 4, 5, 7.*