

COMP9312

Revision & Final Exam

COMP9312_24T2



UNSW
SYDNEY

Review for Graph Algorithms

- Basic data structure to store graphs
 - Advantages/disadvantages of each data structure
- Graph Traversal
 - BFS, DFS, online reachability, connected component detection, topological sort, minimum spanning tree (Prim, Kruskal),
- Reachability
 - Transitive closure, tree cover, two-hop cover
- Path
 - Dijkstra's algorithm, A* search, all-pairs shortest distance/path

Review for Graph Algorithms (Cont)

- Cohesive subgraph detection
 - K-core, k-truss, triangle counting
- Distributed graph processing
 - Mapreduce vs Pregel (why we need graph distributed system), Pregel, Distributed core decomposition, Distributed Connected Component Detection, RDBMS vs Graph Database (why we need graph database)

Learning outcome

We did not introduce too many algorithms in COMP9312, but there are some implementation details you need to consider even only about graph storage and very fundamental algorithms like connected component detection.



When you implement some ideas in the future, whatever programming language you use, think about how we optimize the algorithm. When designing and implementing algorithms, one small step forward may significantly improve efficiency and reduce space usage.

Review for Graph Learning

- Graph Node Feature Engineering
 - Node level/ Edge level feature options
- Node Embedding
 - Encoder/ Decoder framework
- Graph Neural Network
 - GCN, GraphSage, GAT
 - Train GNNs

Final Exam Key Details

- Online exam
- 10am – 1pm (Sydney time) 19th August 2024 (Monday)
- Exam time may be adjusted. **Keep watching your mail box.**
- Released via Webcms (same as the project and assignments)
- Submit answers via Moodle (can submit answers multiple times)
- Do not leave submission until the last minute
- UNSW would not accept any special consideration from people who already attempted the exam
- A sample exam paper can be found on Webcms.

Final Exam Key Details (Cont)

- 8-9 questions (1—2 GNN questions)
- No practical coding work, but pseudocode is definitely required.
- Include several algorithm design questions.
 - Use learned techniques to solve new problems
 - Analyze performance (time complexity, space complexity)
- **Plagiarism checking will be applied.**
- Difficulty? Just try your best to answer questions.
 - The average mark of the final exam will be scaled up to at least **65/100** if it is lower than **65/100**.
 - No scale down applied.

Supplementary Exams

Supplementary Exams are only available to people who are absent from the Final Exam with **good** reason

(good = documented, serious, clearly affects ability to do exam)

If you are awarded a Supp Exam ..

- You must make yourself available for it
- Non-attendance at the Supp => mark of 0 for the exam

Assessment

Of course, assessment isn't a "one-way street" ...

- I get to assess you in the final exam
- you get to assess me in the Course Evaluation

MyExperience evaluations are online (via MyUNSW) NOW

- Telling us good things is ok.
- Telling us things to improve is very useful.

Ask us anything

- Forum.
- Email.
- Consultation in Week 11: K17 G12 Lyre Lab 1pm—2:30pm 8th August (Thursday).
- Discuss your preferred consultation time on the EdForm. We will arrange the next consultation accordingly in week 12.
- Keep watching your email box
 - Final exam details, Project results, Ass2 results...

Other stuffs about Graph

Network X: python graph library

Gephi: graph visualization

Spark GraphX: distributed graph engine

...

AI: get familiar with modern neural networks and models (LLM, GNN, etc.)

DB/Data: practice algorithms and data structure (OJ...)

Beyond this course

Data Structure & Programming & Deep Learning

No single correct answer. (Think about our **project** questions.)

Take pride in your work. (Aim for quality, not just correctness)

PhD scholarships are available, welcome to apply HDR positions if you are interested in this course.

That's all folks!

Good luck with the exams ... and life ...



UNSW
SYDNEY