COMP3821/COMP9801 Workshop Week 1

Graph Reductions and Sweepline Algorithms

Xueyi Chee and Jeremy Le

2025T3

Introductions

Xueyi

- 3rd year Advanced Math / Computer Science
- Took COMP3821 in 24T1
- Likes Crispy Pork Banh Mi
- Dislikes online lectures (recordings too)
- x.chee@unsw.edu.au

Jeremy

- 4th year Computer Science / Mathematics
- Took COMP3821 in 23T1
- Didn't sleep that much :(
- Dislikes classes at 9am...
- jeremy.le1@unsw.edu.au

What is this course about?

- First and foremost, this course is about problem solving and communication!
- We use the tools and techniques taught in lectures and tutorials and apply them to unique and interesting problems.
- Develop intuition to design and thoroughly analyse correct and efficient algorithms.
- Extension to preliminary courses COMP2521/9024 and MATH1081/COMP9020.

Workshop Structure

- Small tutorial part going through guided problems.
- In random groups of 4–6 students, discuss and solve exercises.
- After the workshop, you will grade each team member in your group based on whether they
  - contributed constructively to the discussion, and
  - communicated clearly and respectfully.
- After the workshop (Friday of the week), you will get assigned 1 exercise for which you will need to write down and submit the solution.

Portfolio

- The write-up of the solution to the assigned exercise is due on Friday of the week following the workshop
- Weighted 20% of course mark
- You will receive feedback and a grade from a marker (90% of portfolio mark).
- You will receive a peer grade for your workshop participation (10% of portfolio mark)

2%

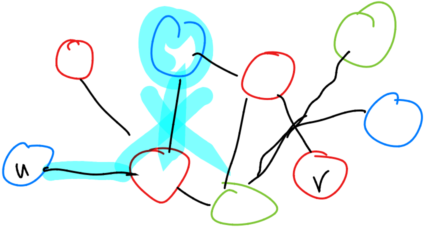
Assessment

- No workshop assessment in Weeks 5, 6 and 10.
- For both the peer mark and the exercise submission, only the best 5 of 7 marks are counted, and averaged to give the portfolio mark.
- Midterm Exam (20%) - 24-hour take home, Friday Week 7.
- Project (20%) - Sundays of Week 3, 7 and 10.
- Final Exam (40%) - 40% hurdle, exam block.
- Up to 5 bonus marks for contributions.

Guided Problem 1

Guided Problem

Let $G = (V, E)$ be an undirected unweighted graph. Each vertex is coloured either `red`, `blue`, or `green`. Given two vertices $u, v \in V$, describe and analyse a polynomial-time algorithm that computes a shortest path from u to v which never visits two consecutive vertices of the same colour.



BFS - unweighted $O(|V|+|E|)$

Dijkstra - handle weighted

negative weighted

Guided Problem 1 Working

① Modifying BFS, add check to the enqueue stage

BFS black box

$$G = (V, E) \longrightarrow G' = (V', E')$$

apply some well known algo

Guided Problem 1 Working

$G' = (V, E')$ $\quad E'$ = E but remove endpoints with the same colour

c color function $\quad c : V \to \{R, G, B\}$

$E' = E \setminus \{(u, v) \in E,\ c(u) = c(v)\}$

Claim ~~St~~ A path in G' $\boxed{\text{corresponds}}$ to a path in G which doesn't visit vertices of same colour.
consecutive

$(\Longleftarrow)$

$(\Longrightarrow)$ Let $P' = v_1 v_2 \dots v_k$ in G'
be a path

Guided Problem 1 Working

$(V_i, V_j) \in E'$ endpoint must have different colours by construction

$\boxed{E' \subseteq E}$ G'

$\therefore$ this gives a path in G that never visits consecutive vertices of same colour

$(\Leftarrow)$ Let P be a path in G

that never visits two consecutive vertices of same colour.

Then it can only pass through different endpoints

Thus must exist in G'.

Guided Problem 1 Working

A path in G

correspondence → same path length

Construction G'

run a BFS from u to find the shortest path to v

construction

$O(|E|)$ $O(|V| + |E|)$ $\Rightarrow$ $O(|V| + |E|)$

Guided Problem 1 Solution Overview

- Construct $G' = (V, E')$ by deleting all edges whose endpoints share the same colour.
- Claim: *Every path in G' corresponds to a path in G such that no pair of consecutive vertices in the path have the same colour*.
- ($\implies$) Let $P = v_1 v_2 \dots v_k$ be a path in G'. Since P uses edges in E', each pair of consecutive vertices in P must have different colours.
- ($\impliedby$) A path in G such that two consecutive vertices of the same colour can only pass through edges with different endpoints.
- Shortest path from u to v in G' corresponds to the shortest path from u to v in G that never visits two consecutive vertices of the same colour.
- Construct G' in $O(|E|)$, then BFS from u to v in $O(|V| + |E|)$ time.

Guided Problem 2

Guided Problem

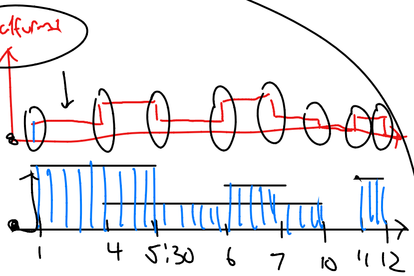
You are given arrival times $A[1..n]$ and departure times $D[1..n]$ of n trains, where the i-th train arrives at time $A[i]$ and departs at time $D[i]$ within a 24-hour period.

Find the minimum number of platforms required so that no two trains occupy the same platform at the same time. Describe and analyse a polynomial-time algorithm to compute this number.

Guided Problem 2 Examples

Example 1:

i	1	2	3	4
A	6:00	4:00	1:00	11:00
D	7:00	10:00	5:30	12:00



Example 2:

i	1	2	3
A	6:00	7:00	8:00
D	10:00	12:00	14:00

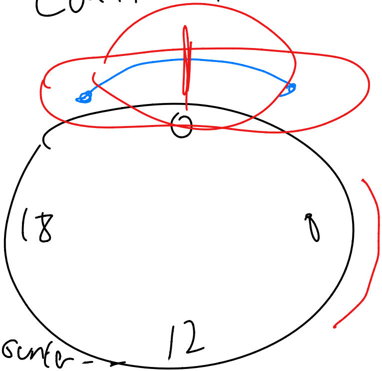
Guided Problem 2 Working

1) Sort all arrivals & departure times in increasing time

1.5) ~~init~~ init a counter

2) loop from earliest time to latest time

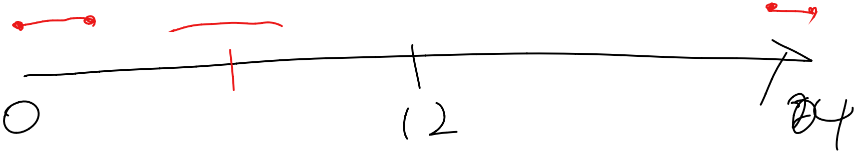
arrival : Counter ++



departure : Counter --

Guided Problem 2 Working

Counter to # . . .



3) return the max val of counter

Guided Problem 2 Working

$X = \max(\text{counter})$

proof:

X is sufficient to accom.. trains

Suppose X is not sufficient

at some time t there were ($Y > X$ trains

at the station

counter would attain Y at some point

$X = \max(\text{counter}) \geq Y > X$

Guided Problem 2 Working

Suppose $\underline{Y \leq X}$ platforms were enough.

at any point in time, number of trains at station $\leq Y$

counter reaches X at some point t

$\exists\, t$ when there are X trains at the station

$\underline{Y \geq X}$ $\qquad X = Y$

Guided Problem 2 Working

$O(n \log n)$

$O(n)$ ...

$O(n \log n)$

$O(n)$

$\Rightarrow O(n \log n)$

Guided Problem 2 Proof

Let X be the maximum value of the counter.

X **platforms can service the train schedule**

- Suppose that X platforms is not sufficient.
- There is some time t where there are $Y > X$ trains at the station.
- By the sweepline algorithm, the counter will have the value Y at some point, so $\max(\text{counter}) \geq Y$.
- So $X = \max(\text{counter}) \geq Y > X$, contradiction!

Guided Problem 2 Proof

X **is the minimum platforms needed**

- Suppose that $Y \leq X$ platforms can support the train schedule.
- At any time t, there is at most Y trains.
- Consider the time t' when the counter attains the value X.
- At t', there are X trains at the station.
- Hence $Y \geq X$.
- So $Y = X$.