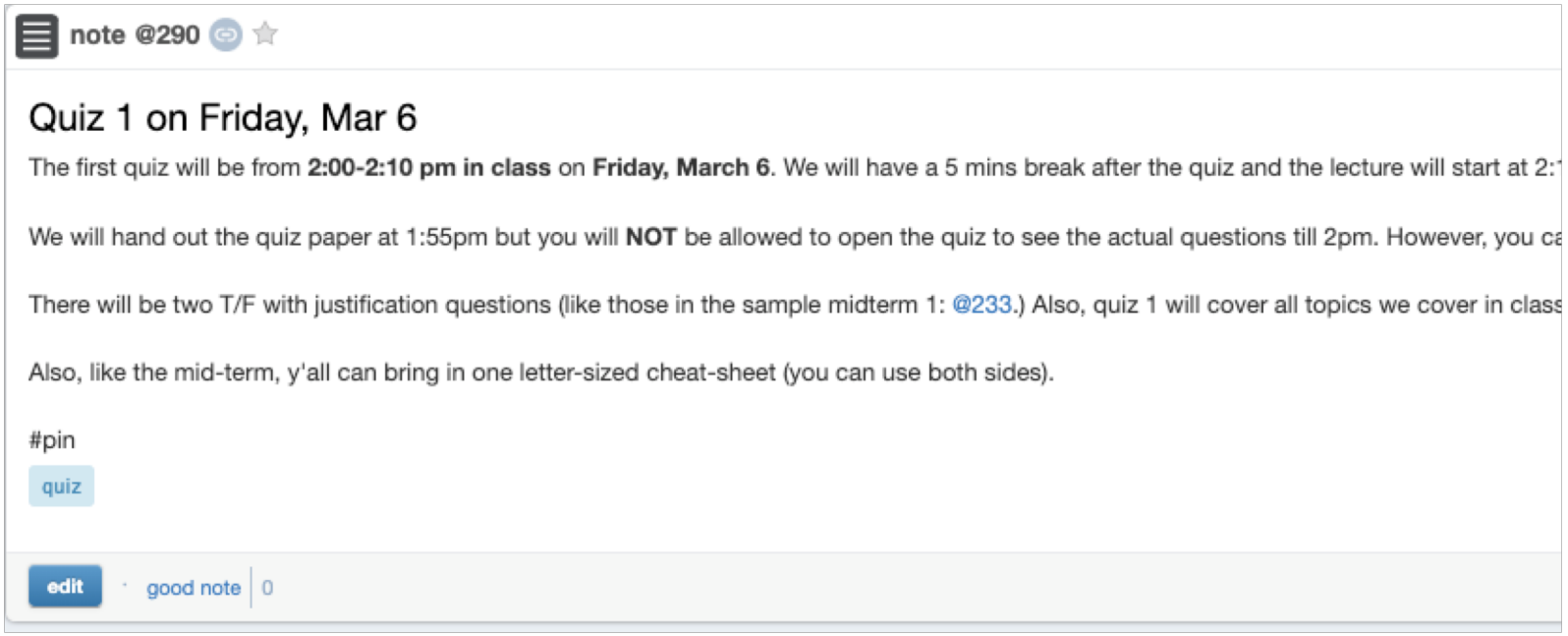
Lecture 16

CSE 331

Mar 2, 2020

Quiz 1 on Friday


note @290

Quiz 1 on Friday, Mar 6

The first quiz will be from **2:00-2:10 pm in class** on **Friday, March 6**. We will have a 5 mins break after the quiz and the lecture will start at 2:

We will hand out the quiz paper at 1:55pm but you will **NOT** be allowed to open the quiz to see the actual questions till 2pm. However, you ca

There will be two T/F with justification questions (like those in the sample midterm 1: @233.) Also, quiz 1 will cover all topics we cover in class

Also, like the mid-term, y'all can bring in one letter-sized cheat-sheet (you can use both sides).

#pin

quiz

edit · good note | 0

Midterms next week: Wed & Fri

Will make a detailed post on Piazza this week.

Teams finalized;
video is due April 10, 11am

CSE 331 Mini project choices

Spring 2020

Please check the table below before submitting your mini project team composition to make sure your case study is not being used by another group. Case studies are assigned on a first come first serve basis.

Group	Chosen Algorithm	Case Study	Links
Tiffany Tate, Joyce Sommer, Robbie Wilkowski (Team TJR)	Predicitve Text Algothrim	Predictive text algorithms are a class of algorithms used to autocomplete/finish words and sentences (e.g. Smart Compose on Gmail).	Link 1, Link 2, Link 3, Link 4
John Tantillo, Joe Brown, Jacob Snyderman (Fingerprinty thingy mabob)	NGI Algorithms with a focus in the AFIT algorithm	NGI algorithms are used for identifying various biometric signatures and matching them to bio signatures on file	Link 1, Link 2, Link 3, Link 4
Steven Jiang, Yang Wenxuan, Steven Quan (od grease)	Facial Recognition Algorithm	Facial Recognition Algorithm is used for Security, Face ID, Camera Focus, Spying	Link 1, Link 2, Link 3, Link 4
Jason Britto, Michael Carlow, Eliza Koster (Codeville)	Pagerank	Pagerank is used to rank webpages on the google search engine	Link 1, Link 2, Link 3, Link 4

Interval Scheduling Problem

Input: n intervals $[s(i), f(i))$ for $1 \leq i \leq n$

Output: A *schedule* S of the n intervals

No two intervals in S conflict

|S| is maximized

Analyzing the algorithm

R: set of requests

Set S to be the empty set

While R is not empty

Choose i in R with the earliest finish time

Add i to S

Remove all requests that conflict with i from R

Return S* = S

S* has no conflicts

S* is an optimal solution

Greedy "stays ahead"

Today's agenda

Prove the correctness

Analyze run-time of the greedy algorithm