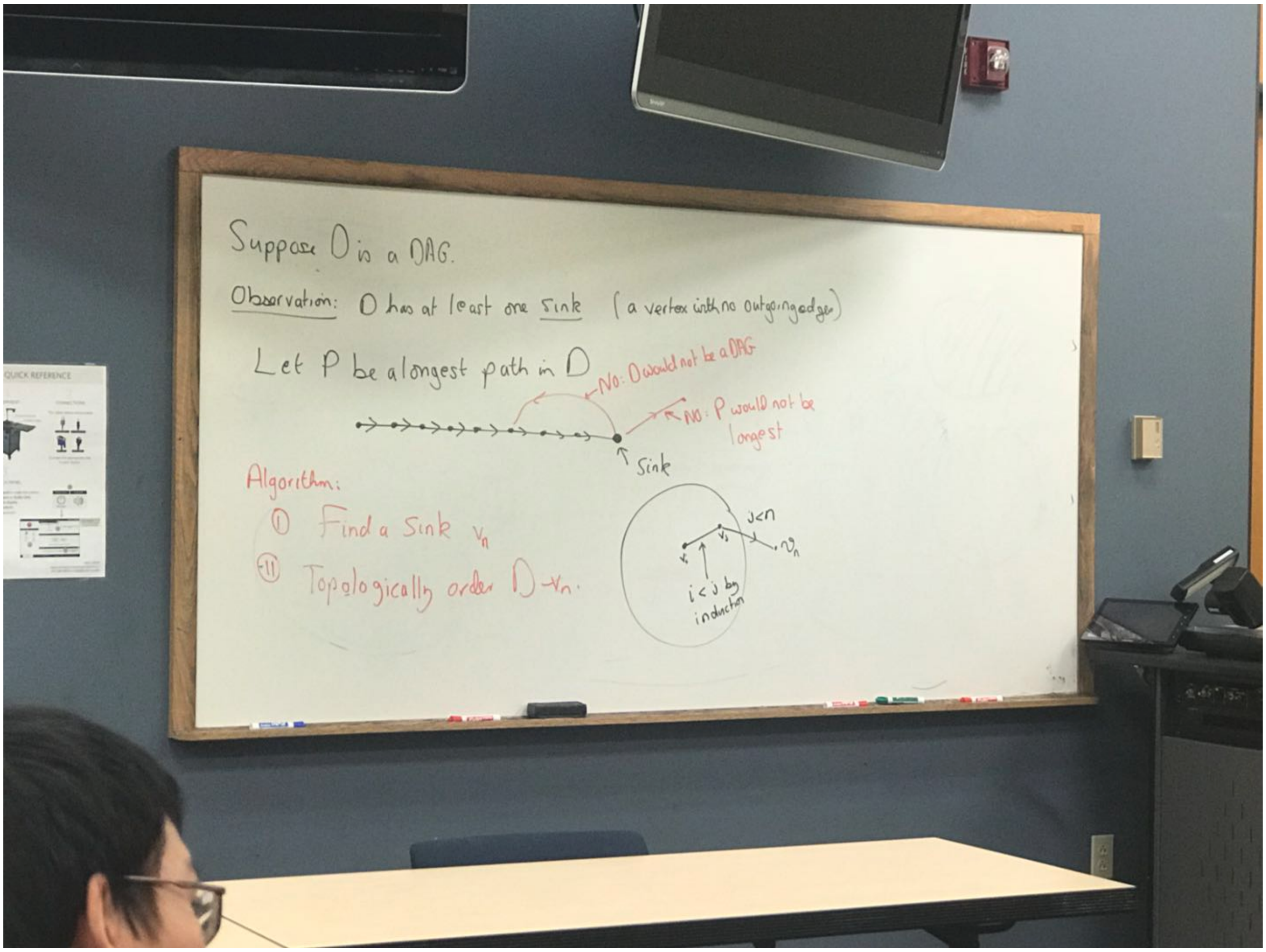
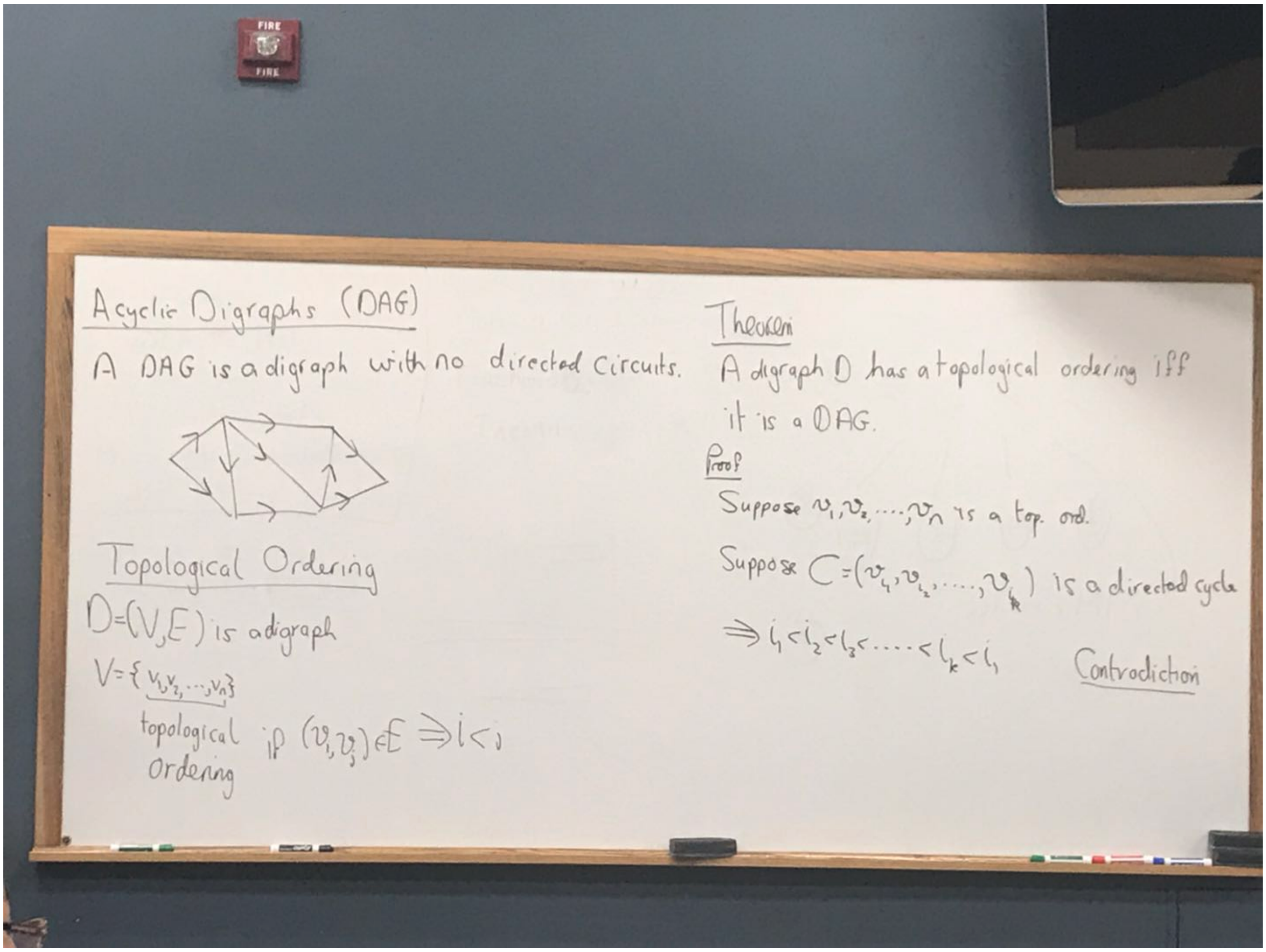
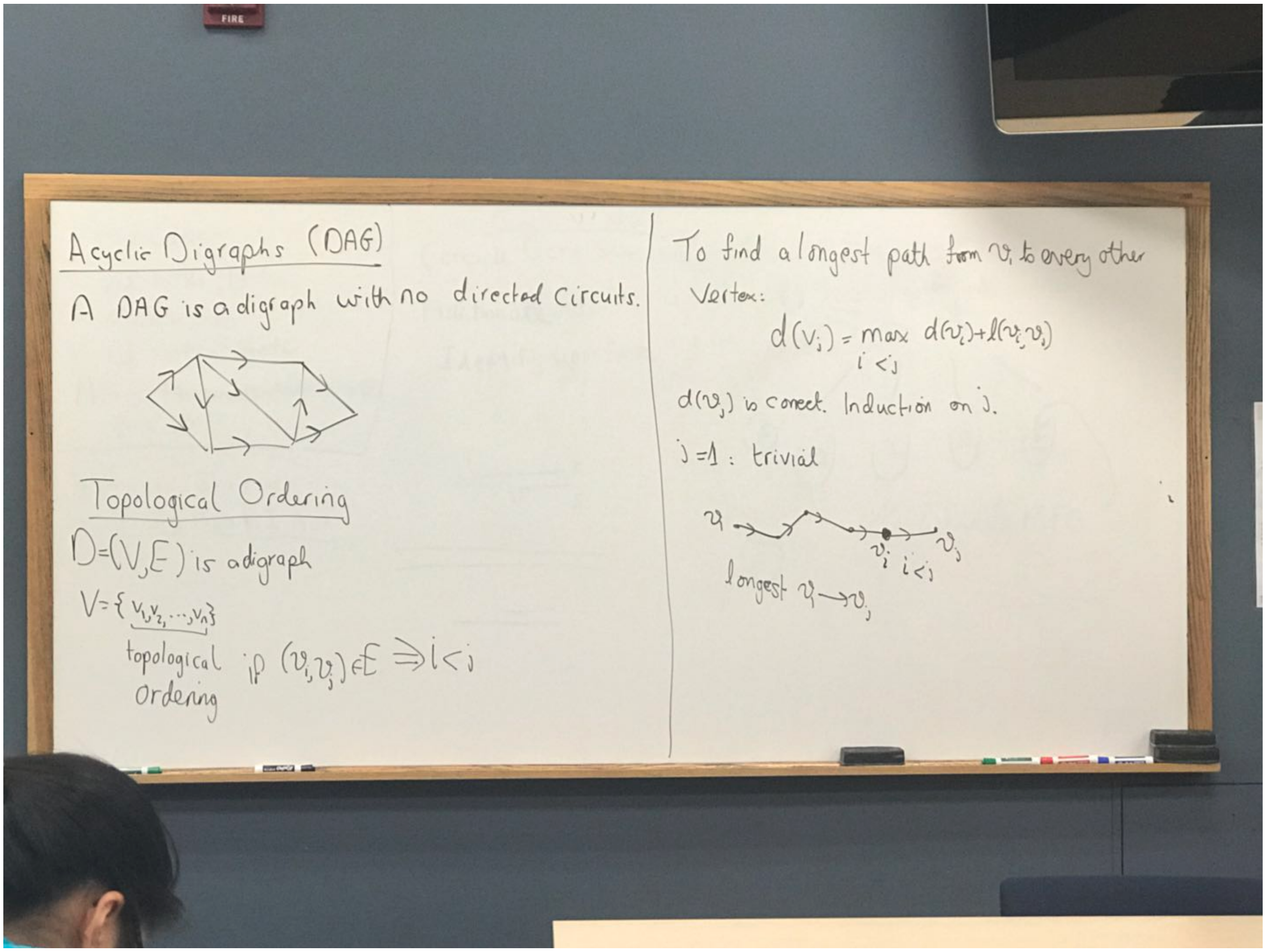
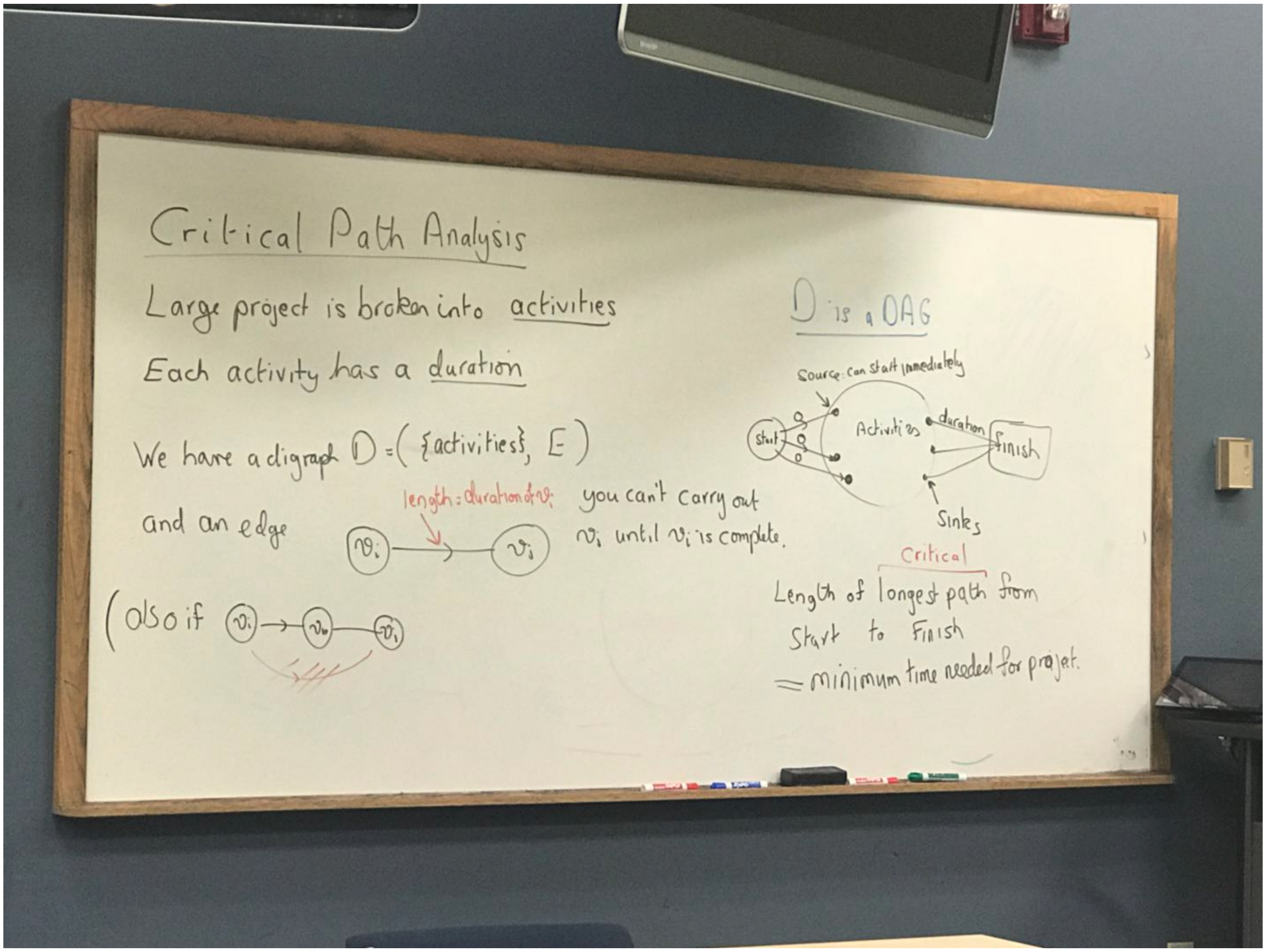
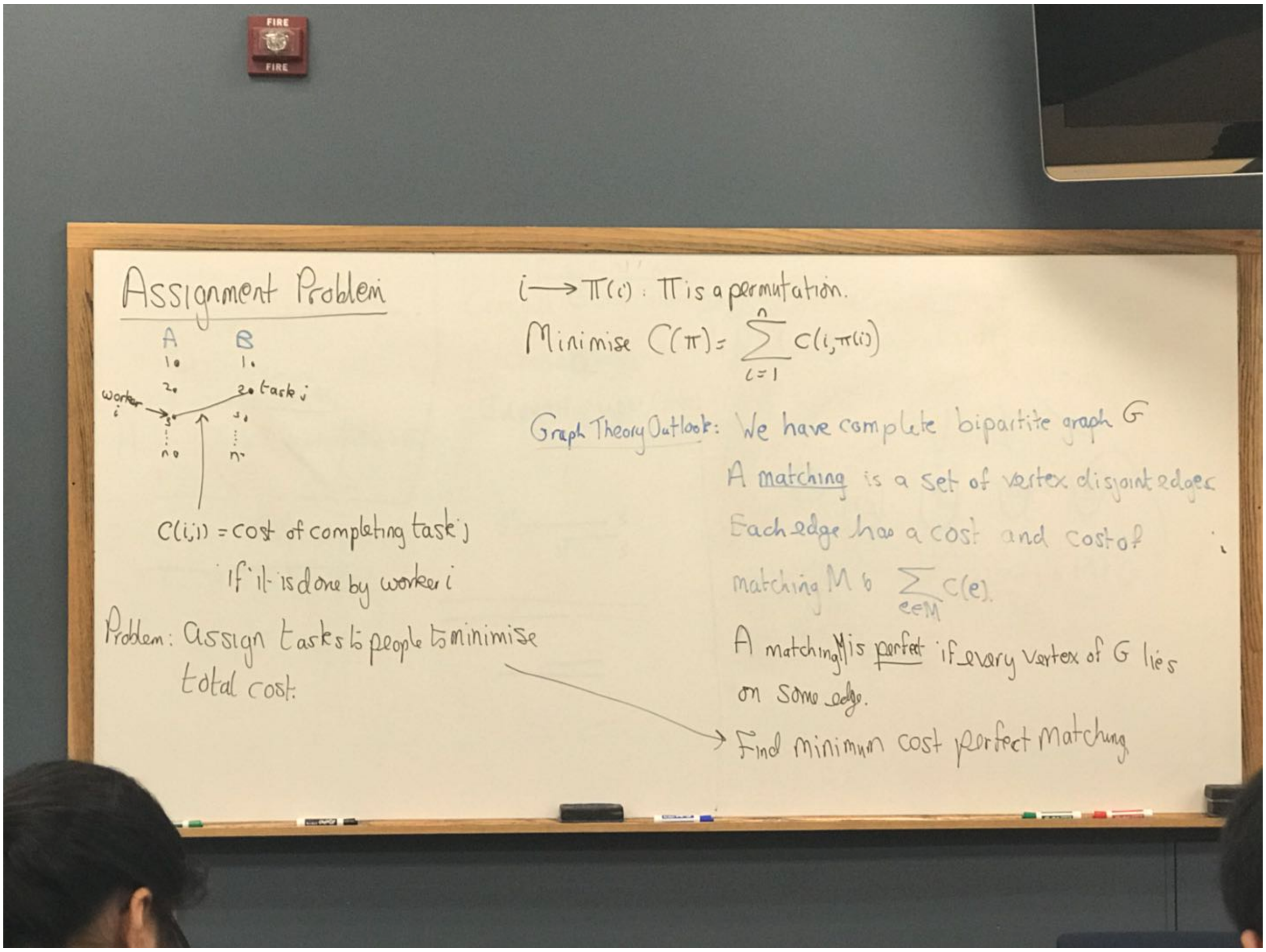
Subject: Fwd: 21-393 Notes September 12
From: Alan Frieze <alan@random.math.cmu.edu>
Date: 09/14/2018 03:07 PM
To: "alan@random.math.cmu.edu" <alan@random.math.cmu.edu>

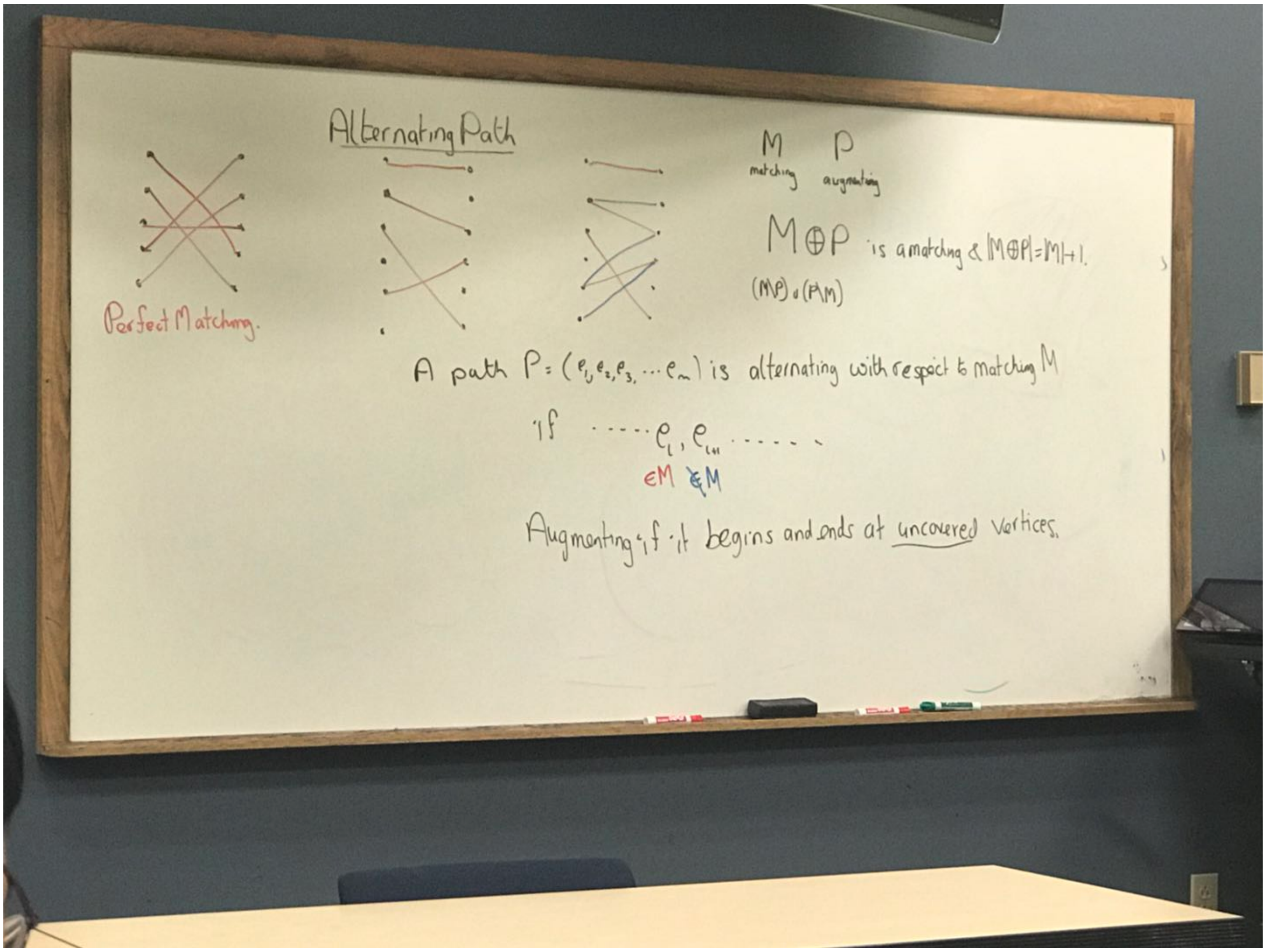



FIRE
Acyclic Digraphs (DAG)
A DAG is a digraph with no directed circuits.
Topological Ordering
D=(V,E) is a digraph
V={v1,v2,...,vn}
topological ordering
if (vi,vj)∈E ⇒ i<j
Theorem
A digraph D has a topological ordering iff it is a DAG.
Proof
Suppose v1,v2,...,vn is a top. ord.
Suppose C=(vi1,vi2,...,vik) is a directed cycle
⇒ i1<i2<i3<....<ik<i1
Contradiction


Acyclic Digraphs (DAG)
A DAG is a digraph with no directed circuits.
Topological Ordering
D=(V,E) is a digraph
V={v1,v2,...,vn}
topological ordering
if (vi,vj)∈E ⇒ i<j
To find a longest path from v1 to every other vertex:
d(vj) = max d(vi)+l(vi,vj)
i<j
d(vj) is correct. Induction on j.
j=1: trivial
v1
vi
vj
i<j
longest v1→vi


Critical Path Analysis
Large project is broken into activities
Each activity has a duration
We have a digraph D = ({activities}, E)
and an edge
length = duration of v_i
you can't carry out
v_j until v_i is complete.
(also if
D is a DAG
Source: can start immediately
Start
Activities
duration
Finish
Sinks
Critical
Length of longest path from
Start to Finish
= minimum time needed for project.


Assignment Problem
A
B
1
2
3
n
worker i
task j
C(i,j) = cost of completing task j
if it is done by worker i
Problem: Assign tasks to people to minimise
total cost.
i ⟶ π(i) : π is a permutation.
Minimise C(π) = Σ_{i=1}^{n} c(i, π(i))
Graph Theory Outlook: We have complete bipartite graph G
A matching is a set of vertex disjoint edges
Each edge has a cost and cost of
matching M is Σ_{e∈M} c(e).
A matching M is perfect if every vertex of G lies
on some edge.
Find minimum cost perfect matching


Alternating Path
Perfect Matching.
M
matching
P
augmenting
M ⊕ P is a matching & |M⊕P|=|M|+1.
(M\P) ∪ (P\M)
A path P = (e1, e2, e3, ... em) is alternating with respect to matching M
if ..... ei, ei+1 ......
∈M ∉M
Augmenting if it begins and ends at uncovered vertices.