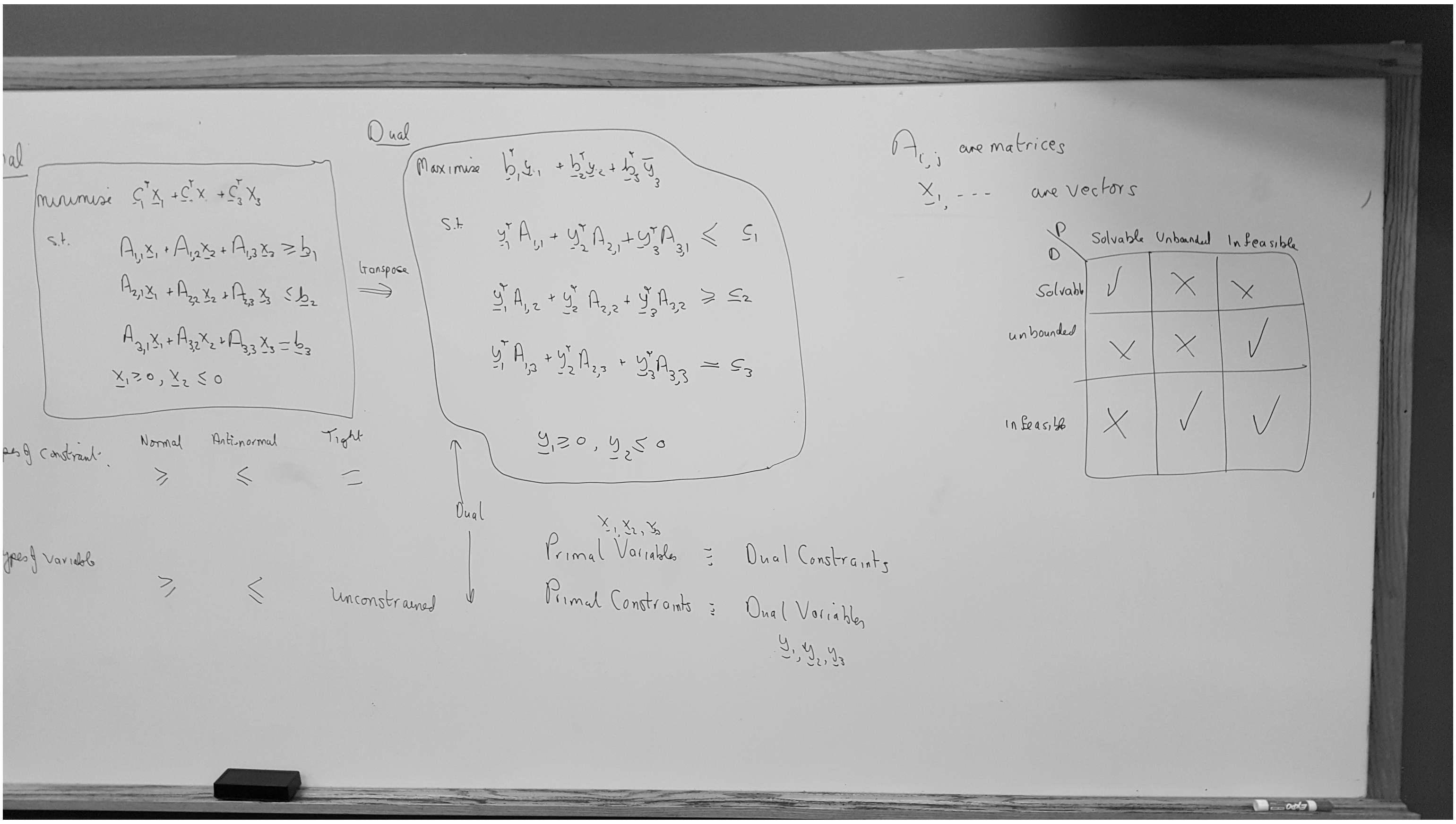
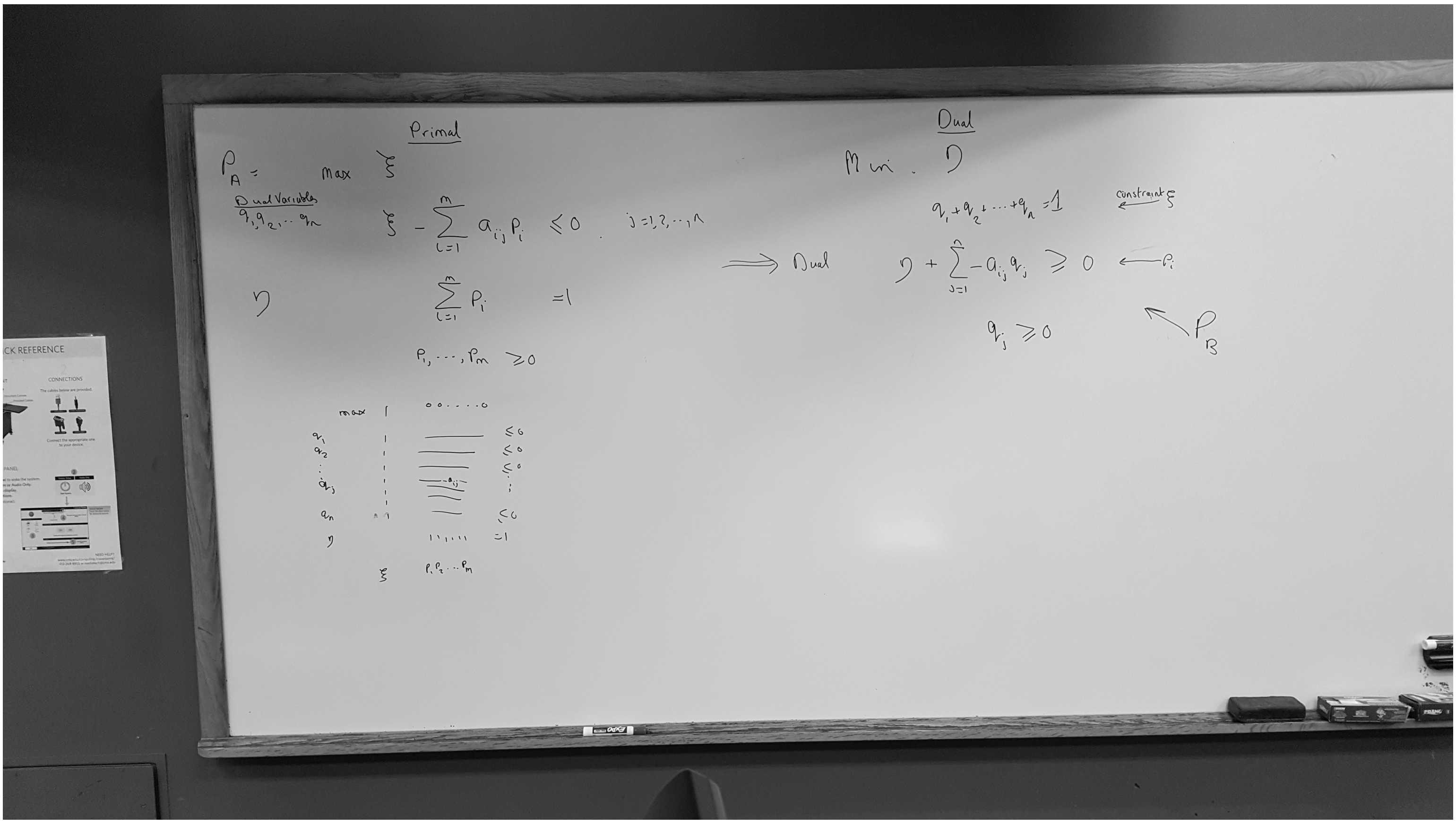


Dual
minimise c1ᵀx1 + c2ᵀx2 + c3ᵀx3
s.t.
A1,1 x1 + A1,2 x2 + A1,3 x3 ≥ b1
A2,1 x1 + A2,2 x2 + A2,3 x3 ≤ b2
A3,1 x1 + A3,2 x2 + A3,3 x3 = b3
x1 ≥ 0, x2 ≤ 0
transpose
Maximise b1ᵀy1 + b2ᵀy2 + b3ᵀy3
s.t.
y1ᵀA1,1 + y2ᵀA2,1 + y3ᵀA3,1 ≤ c1
y1ᵀA1,2 + y2ᵀA2,2 + y3ᵀA3,2 ≥ c2
y1ᵀA1,3 + y2ᵀA2,3 + y3ᵀA3,3 = c3
y1 ≥ 0, y2 ≤ 0
Types of constraint: Normal ≥ Anti-normal ≤ Tight =
Types of variables ≥ ≤ Unconstrained
Dual
x1, x2, x3
Primal Variables ≡ Dual Constraints
Primal Constraints ≡ Dual Variables
y1, y2, y3
Ai,j are matrices
x1, ... are vectors
P
D
Solvable
Unbounded
Infeasible


Primal
ρA = max ξ
Dual Variables
q1, q2, ... qn
ξ − Σ_{i=1}^{m} a_ij p_i ≤ 0 . j = 1, 2, ..., n
η
Σ_{i=1}^{m} p_i = 1
p1, ..., pm ≥ 0
max 1 0 0 0
q1 ≤ 0
q2 ≤ 0
qj a_ij
qn ≤ 0
η 1 1 1 1 1 1 = 1
ξ p1 p2 ... pm
⟹ Dual
Dual
Min. η
q1 + q2 + ... + qn = 1
constraint ξ
η + Σ_{j=1}^{n} −a_ij q_j ≥ 0
p_i
q_j ≥ 0
ρB
QUICK REFERENCE
CONNECTIONS

Simple Aspects

Dominance

$$\begin{bmatrix} 3 & 5 & 7 & 9 & 4 & 6 \\ 2 & 4 & 6 & 9 & 3 & 5 \\ 4 & 6 & 2 & 5 & 3 & 6 \\ 2 & 7 & 5 & 4 & 3 & 3 \end{bmatrix}$$

Row 1 dominates Row 2

Col. 1 dominates Col 6 & Col 2 & Col 4

Now

Row 1 dominates Row 4

$$\downarrow$$

$$\begin{bmatrix} 3 & 7 & 4 \\ 4 & 2 & 3 \end{bmatrix}$$

Non-Singular Game

Suppose A is non-singular and $\underline{1}^T A^{-1} \underline{1} \neq 0$

Value of game is $v = \dfrac{1}{\underline{1}^T A^{-1} \underline{1}}$ $\qquad \underline{1} = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}$

provided

$$\underline{p} = v\, \underline{1}^T A^{-1}, \quad \underline{q} = v A^{-1} \underline{1} \geq 0$$

$\underline{p}$ is feasible for primal

$\underline{q}$ is " " dual

objective values are equal to v.

Symmetric Games

$$A^T = -A$$

Value of game = 0

$$\underline{p}^T A \underline{p} = \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} \, p_i \, p_j = \sum_i a_{ii} p_i + \sum_{i \neq j} (a_{ij} + a_{ji}) p_i p_j$$

$$a_{ii} = -a_{ij} = 0$$

$$a_{ij} + a_{ji} = 0, \; i \neq j$$

PAY($\underline{p}$, $\underline{q}$)

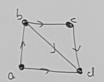
$$\begin{bmatrix} 0 & & & & & & & \\ & 0 & & & & & & \\ & & 0 & & & & & \\ & & & 0 & & & & \\ & & & & 0 & & & \\ & & & & & 0 & & \\ & & & & & & 0 & \\ & & & & & & & 0 \end{bmatrix} \longleftarrow$$

$$P_A \leq 0 \quad \& \quad P_A = P_B \implies P_A = P_B = 0$$

$$P_B \geq 0$$

Shortest Path Problem

A **digraph** $D = (V, A)$ — V: vertices; $A \subseteq V \times V$ are arcs



$V = \{a, b, c, d\}$, $A = \{(b, c), \dots\}$

A walk $W = (\omega_1, \omega_2, \dots, \omega_i, \omega_{i+1}, \dots, \omega_N)$ is a sequence of vertices such $(\omega_i, \omega_{i+1}) \in A$, $\forall i$

W is closed if $\omega_1 = \omega_N$.

Trail: no edge used more than once.

Path: no vertex used more than once.

Suppose each arc $e \in A$ has a length $l(e)$.

If $P = (v_1, v_2, \dots, v_k)$ then we can extend l to paths.

$$l(P) = l(v_1, v_2) + l(v_2, v_3) + \dots + l(v_{k-1}, v_k).$$

Shortest Path Problem: find minimum length paths between pairs of vertices.

Assume for now that $l(e) \geq 0$, $\forall e \in A$.

Single Source Shortest Path Problem:

Given $s \in V$, find a shortest path from s to each $v \in A$.

Dijkstra's Algorithm

Step 0 $d(s) = 0$ & $d(v) = \infty$, $v \neq s$

$A = \{s\}$ and $B = V \setminus \{s\}$

$i = 1$, $v_1 = s$

[At any stage of the algorithm, if $v \in A$ then $d(v)$ = length of shortest path from s to v

if $v \in B$ then $d(v)$ is an estimate.

Step 1

If $v \in B$ let $d(v) \leftarrow \min\{d(v), d(v_i) + l(v_i, v)\}$ (*)

Step 2 Let $d(v^*) = \min_{v \in B} d(v)$.

Claim: $d(v^*)$ is correct

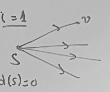
$A \leftarrow A \cup \{v^*\}$

$B \leftarrow B \setminus \{v^*\}$

$i \leftarrow i+1$; $v_{i+1} = v^*$

(return to Step 1)

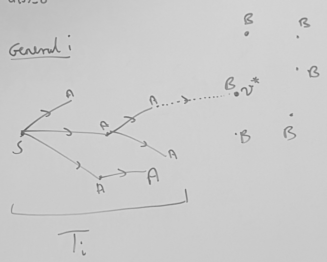
$i = 1$



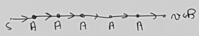
$d(v) = l(s,v)$

$d(s) = 0$

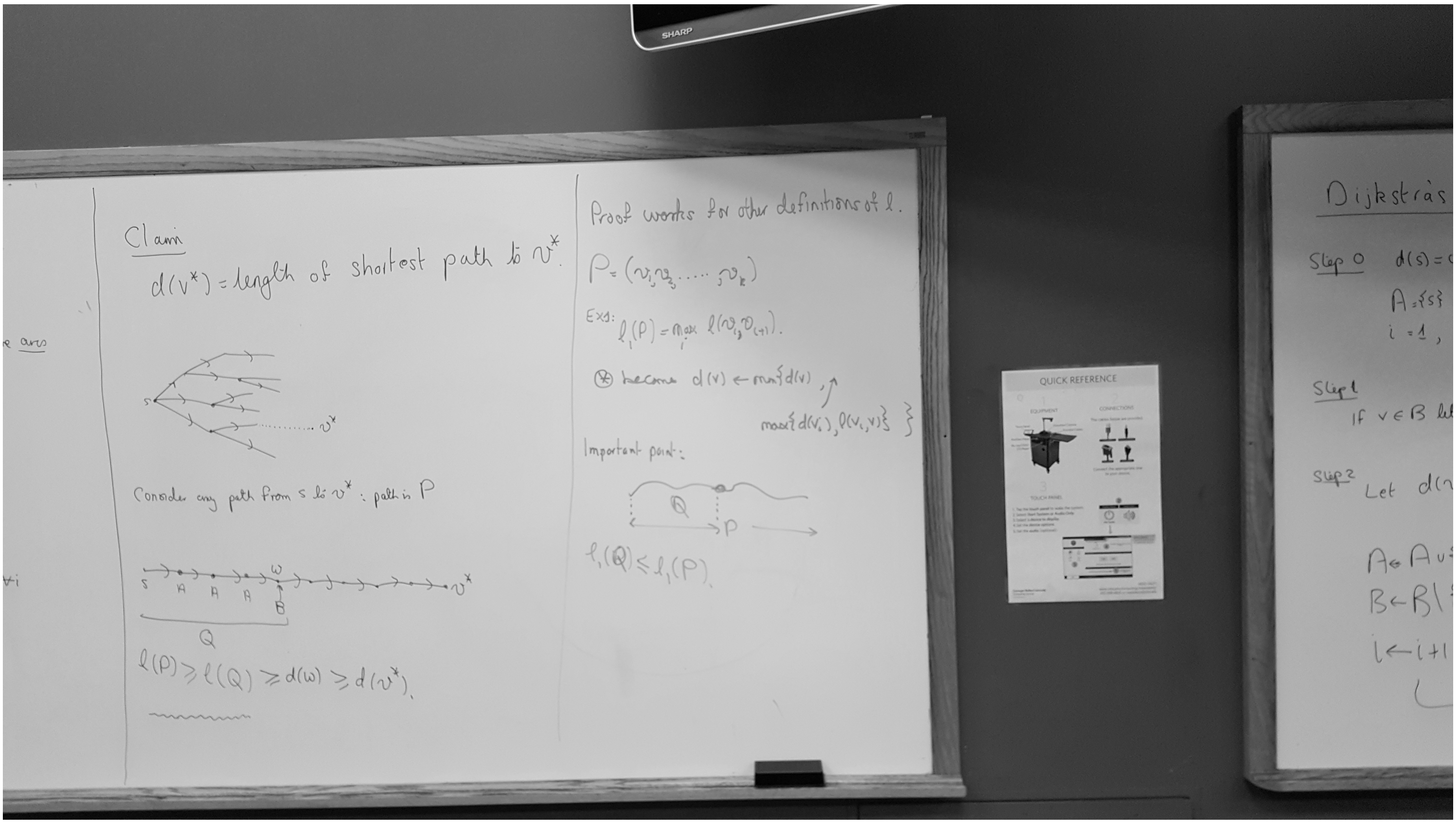
General i

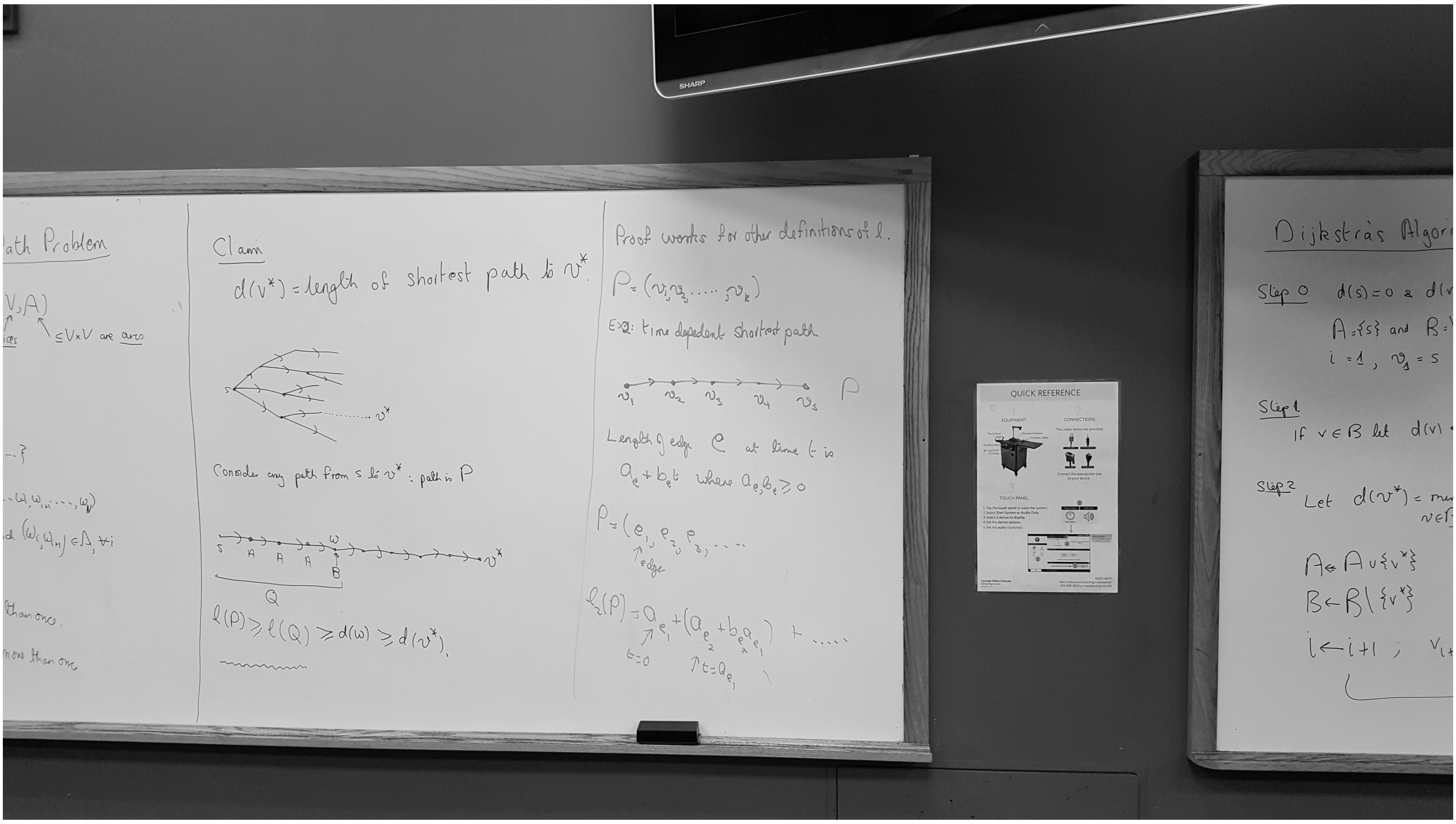


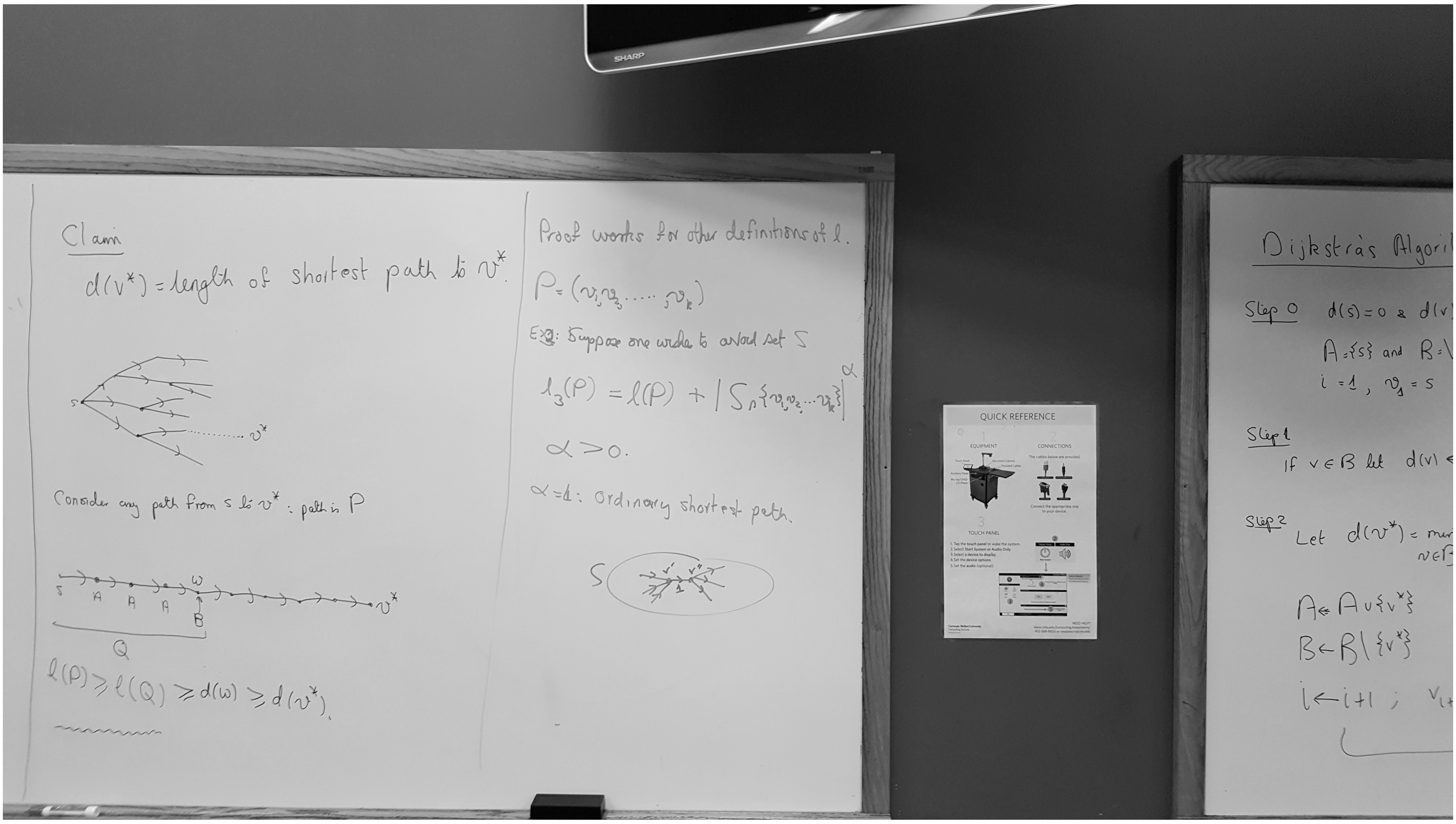
Observation: if $v \in B$ then $d(v)$ is the minimum length path from s to v of the form



(*) just updates this minimum by checking one new path to v.


SHARP
Claim
d(v*) = length of shortest path to v*.
Consider any path from s to v*: path is P
s A A A w B v*
Q
l(P) ≥ l(Q) ≥ d(w) ≥ d(v*).
Proof works for other definitions of l.
P = (v1, v2, ..., vk)
Ex1: l1(P) = max l(vi, vi+1).
Important point:
Q
P
l1(Q) ≤ l1(P).
QUICK REFERENCE
Dijkstra's
Step 0
A = {s}
Step 1
Step 2
Let


SHARP
Path Problem
Claim
d(v*) = length of shortest path to v*.
Consider any path from s to v* : path is P
ℓ(P) ≥ ℓ(Q) ≥ d(w) ≥ d(v*).
Proof works for other definitions of ℓ.
Ex: time dependent shortest path
Length of edge e at time t is
a_e + b_e t where a_e, b_e ≥ 0
QUICK REFERENCE
Dijkstra's Algor
Step 0
Step 1
Step 2


Claim
d(v*) = length of shortest path to v*.
Consider any path from s to v* : path is P
Q
l(P) ≥ l(Q) ≥ d(w) ≥ d(v*).
Proof works for other definitions of l.
P = (v1, v2, ..., vk)
Ex: Suppose one wishes to avoid set S
α > 0.
α = 1 : Ordinary shortest path.
S
QUICK REFERENCE
EQUIPMENT
CONNECTIONS
TOUCH PANEL
Dijkstra's Algori
Step 0
A = {s} and
i = 1, v1 = s
Step 1
If v ∈ B let
Step 2
Let
A ← A ∪ {v*}
B ← B \ {v*}
i ← i+1 ;
SHARP