

Video Lectures On Artificial Intelligence

Lecture 10 Hill Climbing

Prof. Deepak Khemani

Department of Computer Science and Engineering
IIT Madras



Heuristic Search — Best First Search

Heuristic Search — Best First Search
heuristic fn $h(n)$



Best First Search
heuristic fn. $h(n)$

$OPEN \leftarrow \text{sort}_h(\text{append New Tail}(OPEN))$

Best First Search
heuristic fn $h(n)$

$nodePair \leftarrow Head(OPEN)$

$OPEN \leftarrow sort_h(append\ New\ Tail(OPEN))$

Search

— Best First Search
heuristic fn $h(n)$

nodePair \leftarrow Head(OPEN)
⋮

PRIORITY
QUEUE \leftarrow OPEN

\leftarrow sort_h(append New T

Heuristic Search — Best First Search
heuristic fn. $h(n)$

nodePair

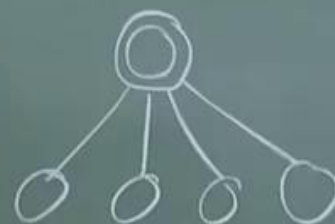
PRIORITY-
QUEUE ← OPEN ←

Heuristic Search

— Best First Search

heuristic fn. $h(n)$

nodePair



PRIORITY-
QUEUE

← OPEN ←

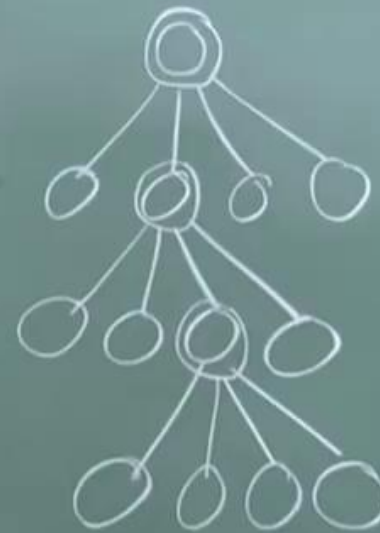


NPTEL

Heuristic Search

— Best First Search:
heuristic fn. $h(n)$

nodePair



PRIORITY-
QUEUE

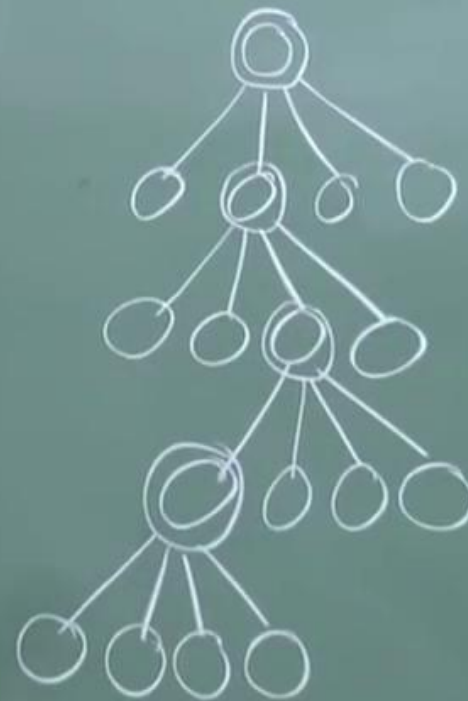
← OPEN ←

Heuristic Search — Best First Search
heuristic fn $h(n)$

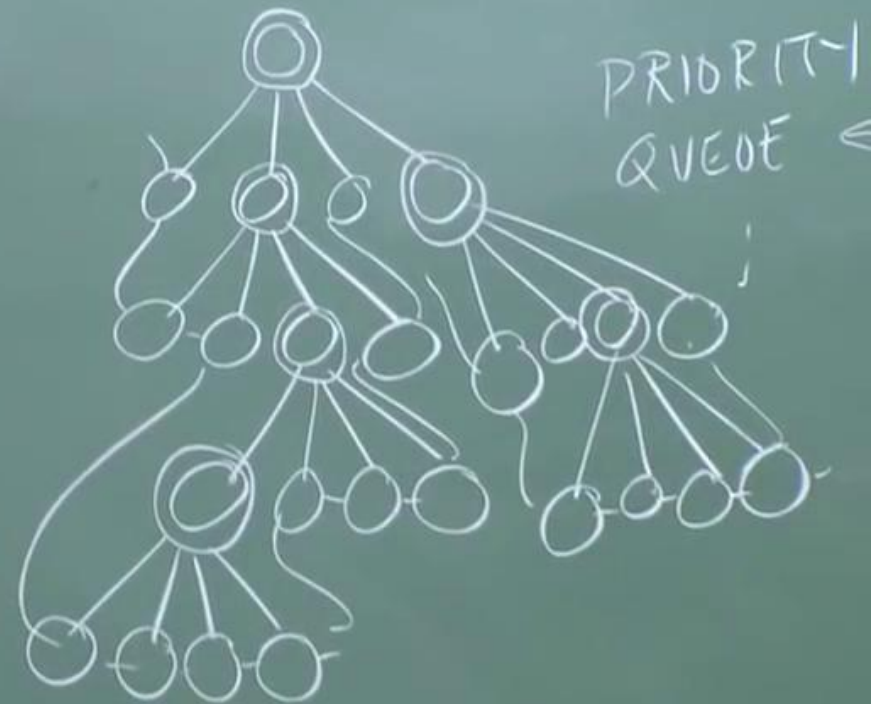
nodePair

PRIORITY-
QUEUE

OPEN



Heuristic Search — Best First Search
heuristic fn $h(n)$



Heuristic Search — Best First Search

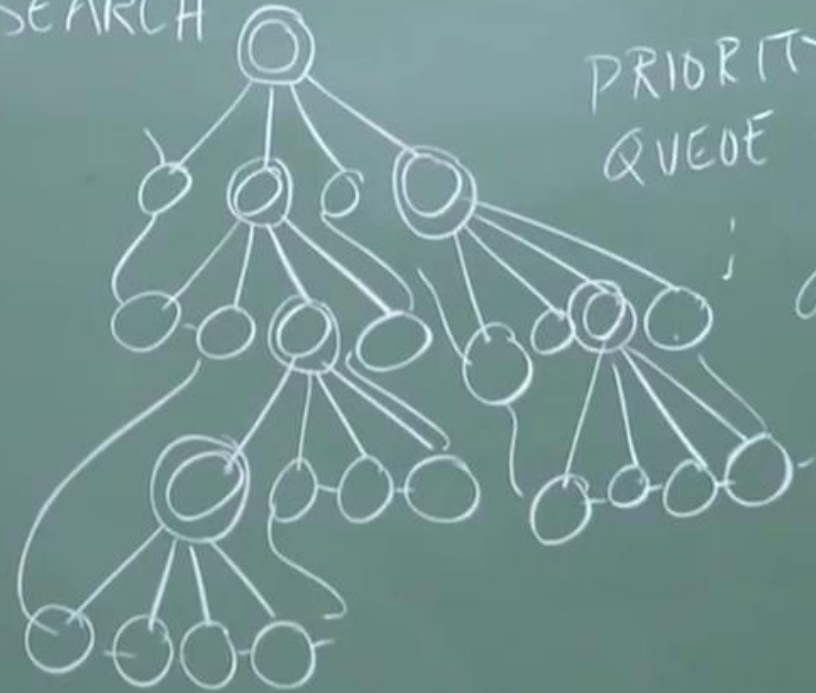
heuristic fn $h(n)$

GLOBAL SEARCH

PRIORITY
QUEUE

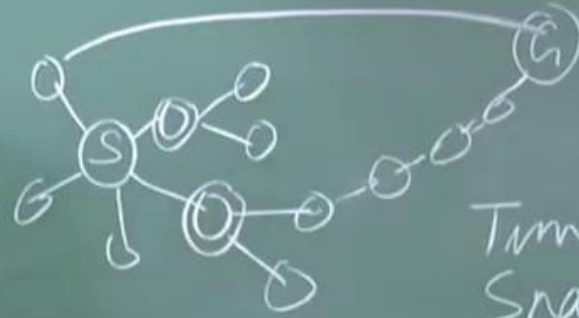
OPEN

GLOBAL



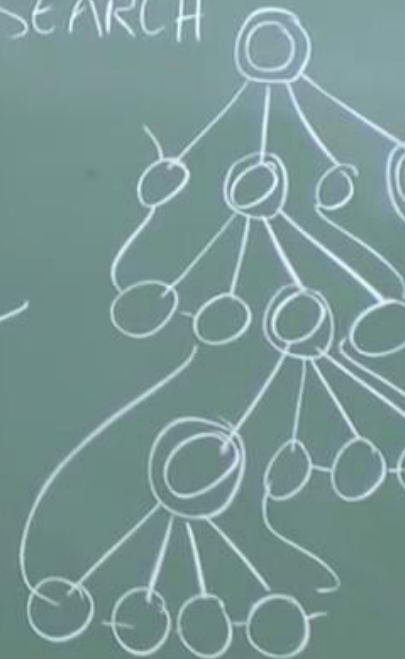
Time
Space
Quality
Completeness

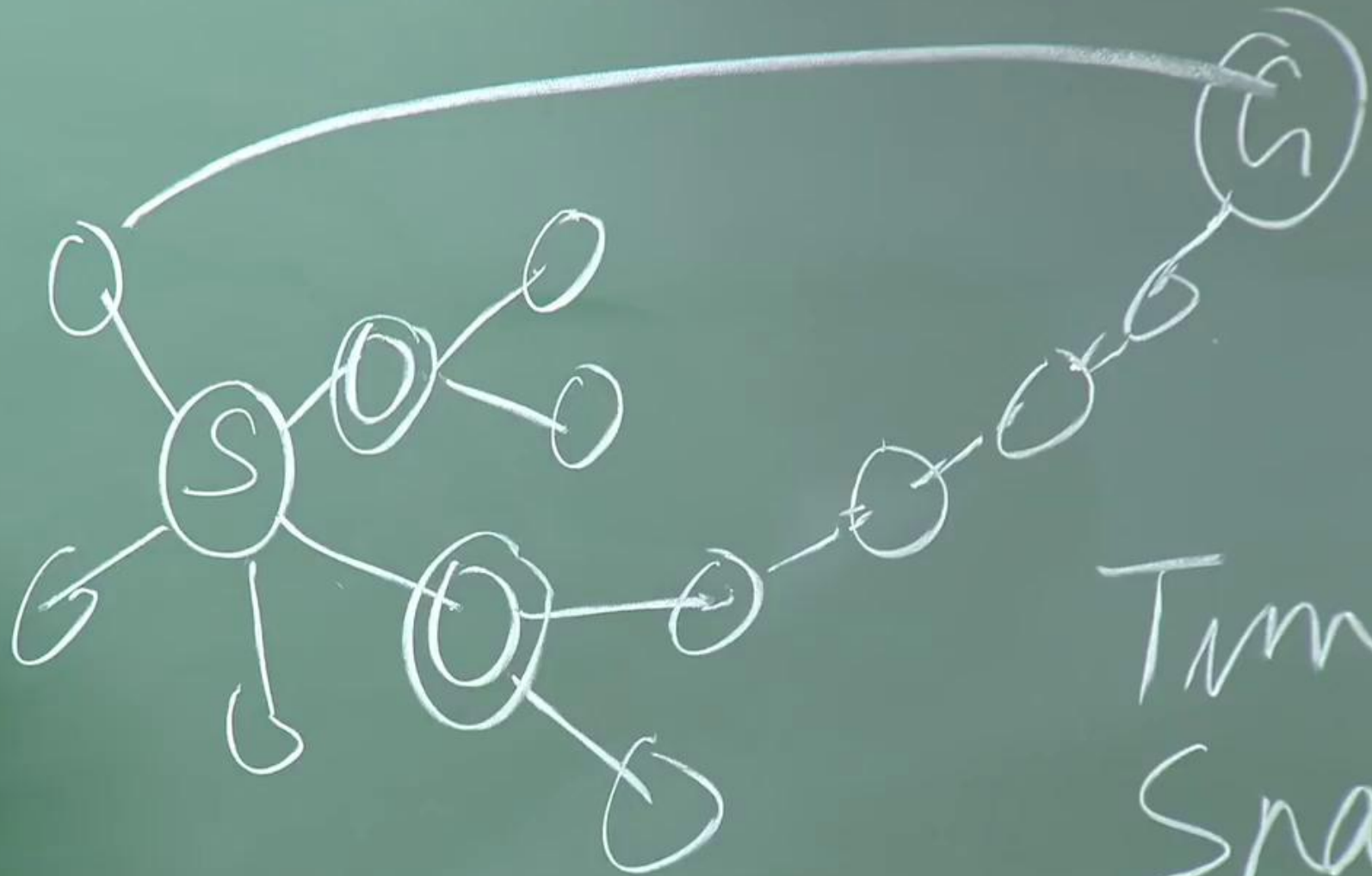
Heuristic Search — Best First heuristic function



GLOBAL SEARCH

Time
Space
Quality ✓
Completeness ✓





Time
Space
Complexity

Heuristic Search — Best First Search

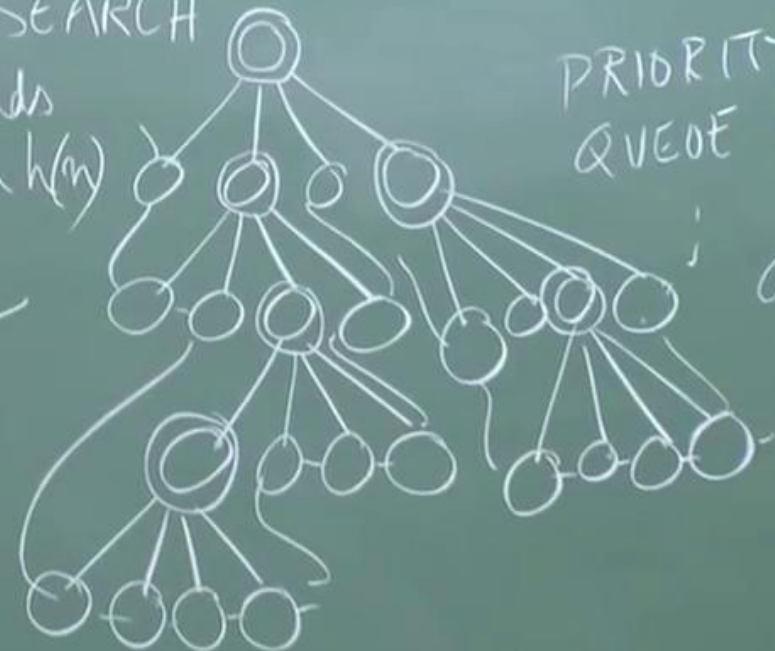
heuristic fn $h(n)$

GLOBAL SEARCH



Time Space
Quality \times
Completeness \checkmark

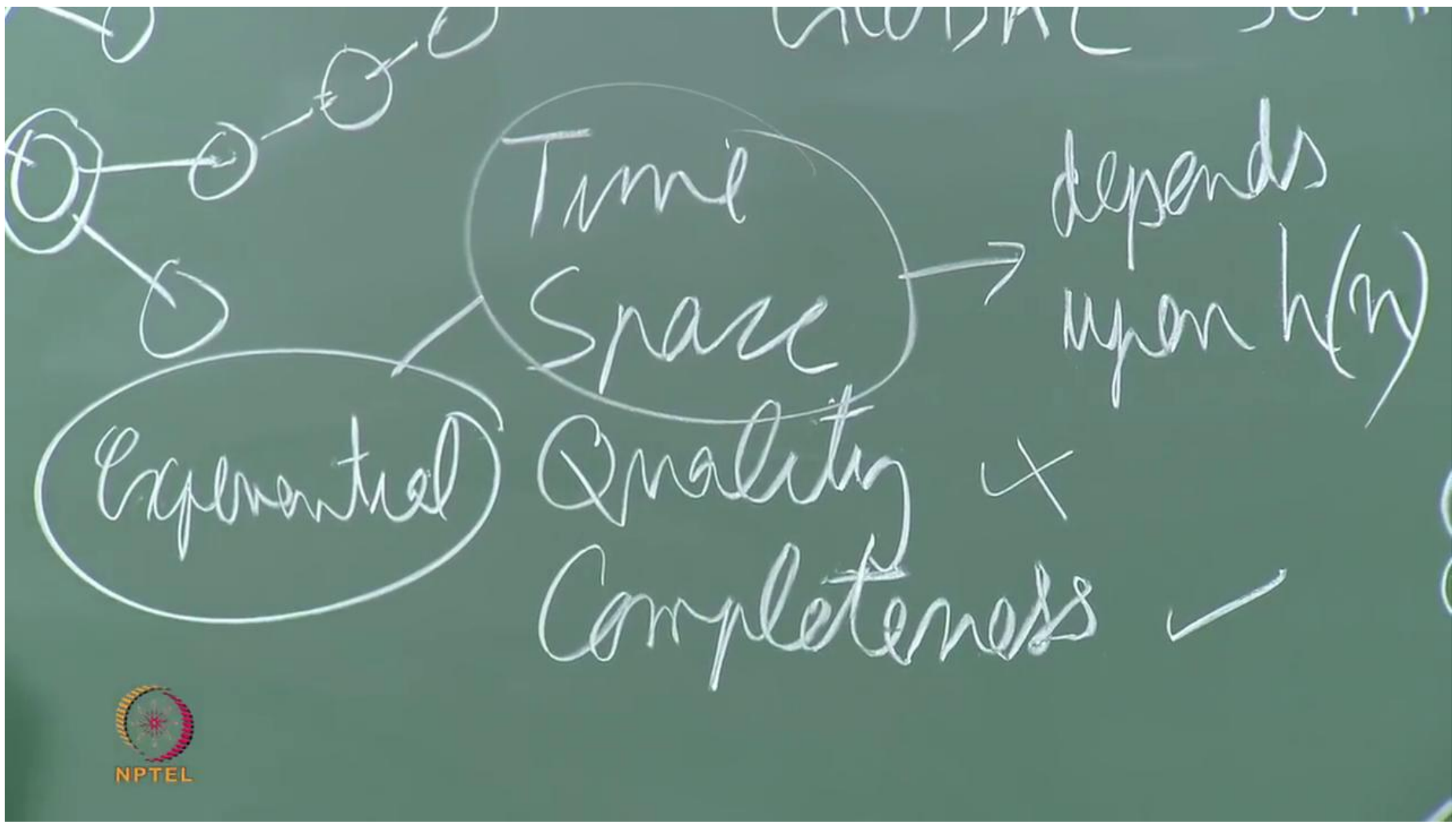
depends upon $h(n)$



PRIORITY QUEUE \leftarrow



NPTEL

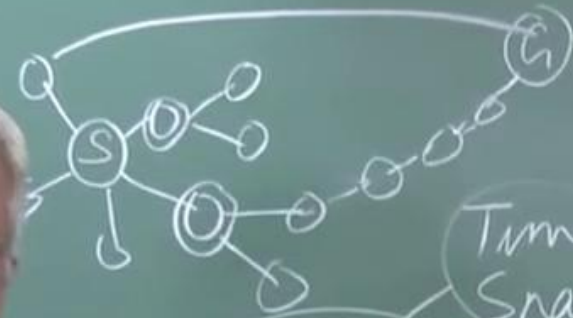




OPEN \leftarrow sort_h(New)

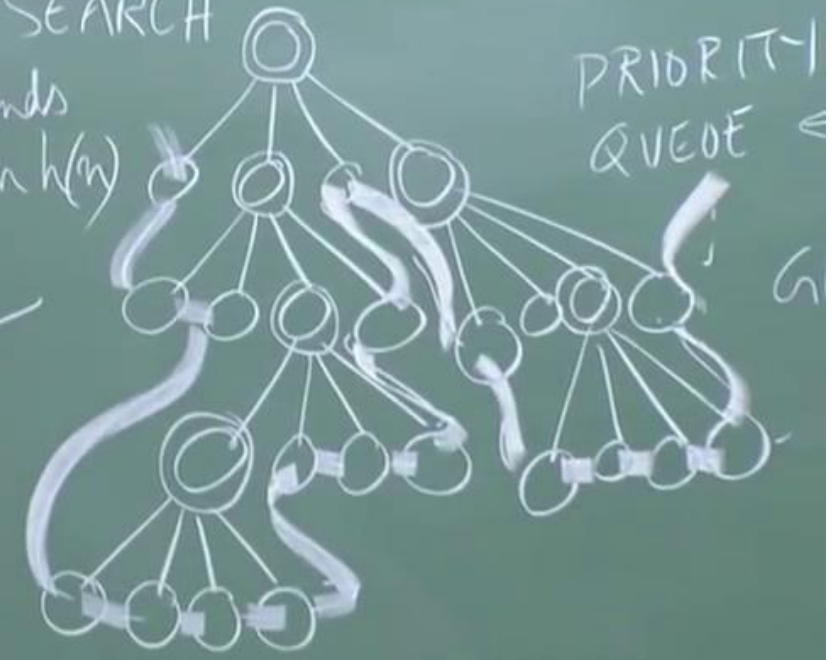
Heuristic Search — Best First Search

heuristic fn $h(n)$



GLOBAL SEARCH

Time Space → depends upon $h(n)$
Exponential Quality × Completeness ✓



Search — Best First Search:
heuristic fn $h(n)$

GLOBAL SEARCH

depends upon $h(n)$
optimality ✓
completeness ✓



nodePair \leftarrow Head(OPEN)
⋮

← OPEN ← $\text{sort}_h(\text{append New Tail})$
GLOBAL

⇓
OPEN $\leftarrow \text{sort}_h(\text{New})$

First Search
fn $h(n)$

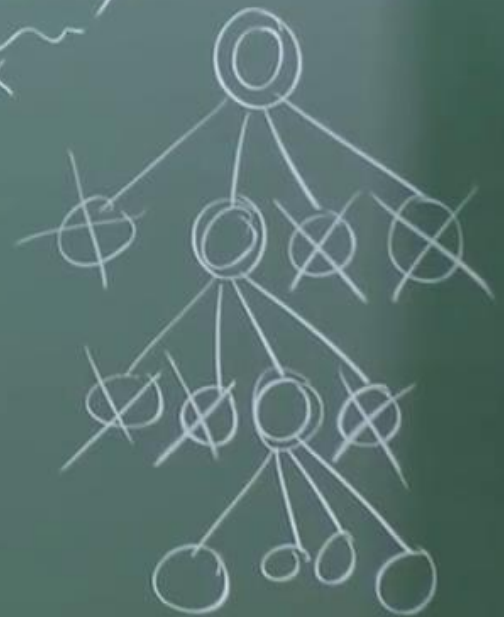
nodePair \leftarrow Head(OPEN)

PRIORITY-
QUEUE \leftarrow

$\text{sort}_h(\text{append New Tail(OPEN)})$



OPEN $\leftarrow \text{sort}_h(\text{New})$



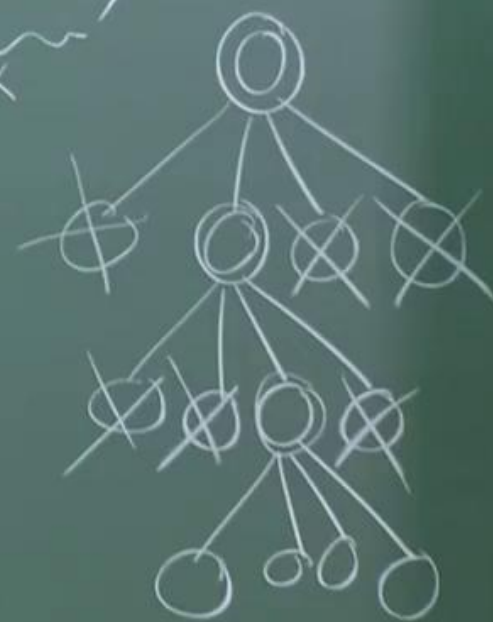
First Search
fn $h(n)$

$$\text{nodePair} \leftarrow \text{Head}(\text{OPEN})$$
$$\text{OPEN} \leftarrow \text{sort}_h(\text{append New Tail}(\text{OPEN}))$$

GLOBAL

$$\text{OPEN} \leftarrow \text{sort}_h(\text{New})$$

Next \leftarrow Best (moveGen(Current))



t First Search
fn $h(n)$

nodePair \leftarrow Head(OPEN)

\vdots
 \leftarrow sort_h(append New Tail(OPEN))



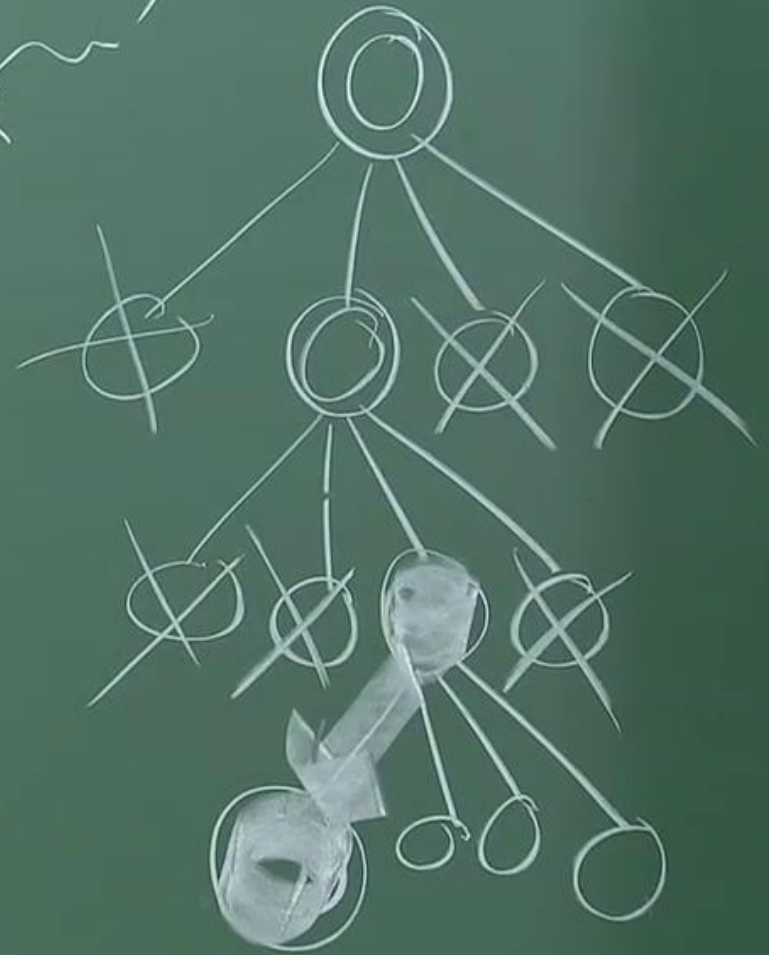
OPEN \leftarrow sort_h(New)

Next \leftarrow Best(moveGen(Current))

PRIORITY-
QUEUE



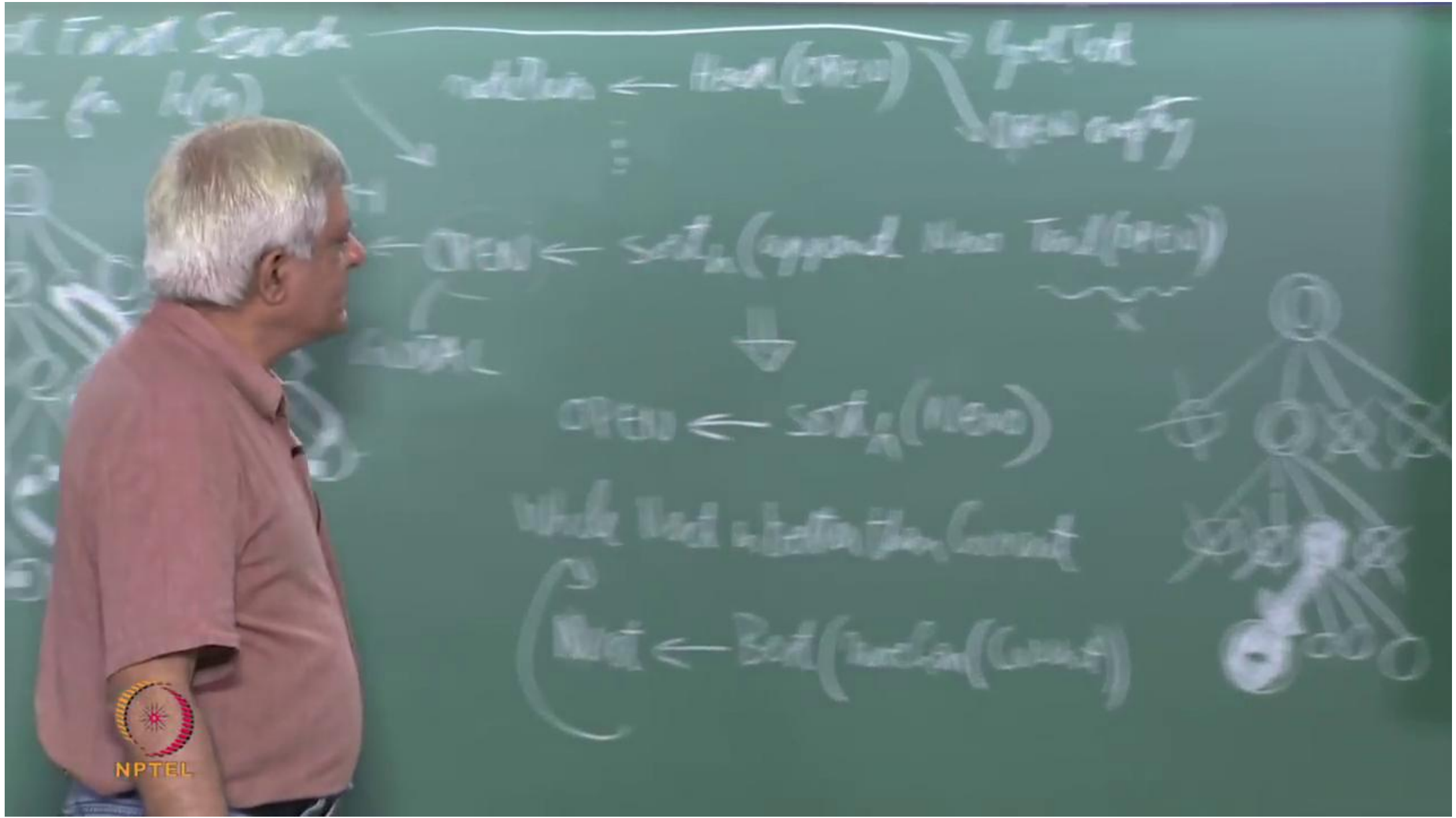
Best \leftarrow Best (moveGen (Current))



$OPEN \leftarrow \text{sort}_h(\text{New})$

While Next is better than Current

$\text{Next} \leftarrow \text{Best}(\text{moveGen}(\text{Current}))$

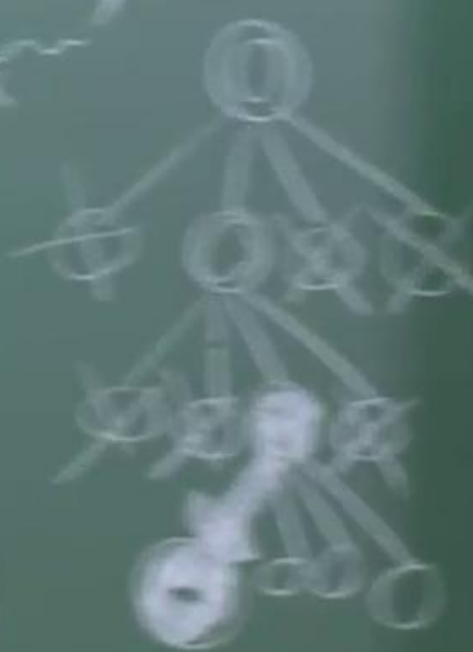


Breadth First Search
for $h(n)$

$node_in \leftarrow \text{Head}(\text{OPEN})$
...
if $node_in$ is goal then return
else $\text{OPEN} \leftarrow \text{set}_A(\text{append}(\text{New_Child}(node_in), \text{OPEN}))$

$\text{OPEN} \leftarrow \text{set}_A(\text{append}(\text{New_Child}(node_in), \text{OPEN}))$
↓
 $\text{OPEN} \leftarrow \text{set}_A(\text{New})$

While Node is not null then
(
 $Next \leftarrow \text{Next}(\text{Node})$
 $Node \leftarrow \text{Next}$
)



Best First Search

heuristic fn $h(n)$

SEARCH



PRIORITY-
QUEUE

nodePair \leftarrow Head(OPEN) $\left\{ \begin{array}{l} \text{Goal Test} \\ \text{OPEN empty} \end{array} \right.$

$\text{sort}_h(\text{append New Tail(OPEN)})$



$\leftarrow \text{sort}_h(\text{New})$

is better than Current

Best (move Gen (Current))



Best First Search

heuristic fn $h(n)$

nodePair \leftarrow Head(OPEN)

Goal Test

OPEN empty

PRIORITY-QUEUE \leftarrow OPEN

GLOBAL

sort_h(append New Tail(OPEN))



OPEN \leftarrow sort_h(New)

While Next is better than Current

Next \leftarrow Best (moveGen(Current))

depends upon $h(n)$

mess



While Next is better than Current
Next \leftarrow Best (move Gen (Current))

construct fn $h(n)$

H

P

$OPEN \leftarrow sort_h(\text{append New Tail}(OPEN))$

\Downarrow

$OPEN \leftarrow sort_h(New)$

While Next is better than Current

Next $\leftarrow Best(moveGen(Current))$

\Downarrow

OPTIMIZATION - Optimize $h(n)$

GLOBAL

OPEN empty

NPTEL



A stick figure is drawn on the left side of the chalkboard, standing on a diagonal line that slopes upwards from left to right. The figure is holding a ball in its right hand. The line represents the direction of the steepest gradient ascent.

STEEPEST GRADIENT ASCENT



www
N

STEEPEST GRADIENT ASCENT

heuristic fn $h(n)$

GLOBAL SEARCH

depends upon $h(n)$

PRIORITY QUEUE \leftarrow OPEN $\leftarrow \text{sort}_h(a)$

GLOBAL

HILL CLIMBING

While Next mb

Next \leftarrow Be

OPTI

STEEPEST GRADIENT ASCENT

The image shows a man in a pink shirt pointing at a chalkboard. The chalkboard contains several diagrams and text related to search algorithms. At the top, 'heuristic fn $h(n)$ ' is written. Below it, 'GLOBAL SEARCH' is written. A tree diagram is drawn with nodes and edges, and a path is highlighted with a thick white line. To the right of the tree, 'PRIORITY QUEUE' and 'OPEN' are written, with arrows indicating a flow. Further right, 'sort_h(a)' is written. Below the tree, 'GLOBAL' is written. On the right side of the board, 'HILL CLIMBING' is written, followed by 'While Next mb' and 'Next \leftarrow Be'. At the bottom right, 'OPTI' is written. At the bottom center, 'STEEPEST GRADIENT ASCENT' is written. The man is pointing at a stick figure drawn on the board, which is positioned near the 'STEEPEST GRADIENT ASCENT' text.

heuristic fn $h(n)$

GLOBAL SEARCH

Time Space → depends upon $h(n)$

Quality × Completeness ✓

PRIORITY QUEUE ← OPEN ← SOL

GLOBAL

HILL CLIMB

While Next

Next ←

STEEPEST GRADIENT ASCENT

The chalkboard contains several diagrams and text. At the top left, a small tree diagram shows a root node 'S' with three children. To its right, a larger tree diagram illustrates a search process, with a path highlighted by thick white lines. To the right of this tree is a diagram of a 'PRIORITY QUEUE' with an 'OPEN' node and an arrow pointing to 'SOL'. Below the tree, a stick figure is shown climbing a wavy line labeled 'STEEPEST GRADIENT ASCENT'. On the right side, a box labeled 'HILL CLIMB' contains the text 'While Next' and 'Next ←'. The man is pointing towards the tree diagram.

hemispheric fn. $h(n)$

GLOBAL SEARCH

Time Space → depends upon $h(n)$

Quality × Completeness

PRIORITY QUEUE ← OPEN ← S

GLOBAL

HILL CLIMBING

While []

Next

STEEPEST GRADIENT ASCENT

The image shows a man with grey hair, wearing a maroon shirt and blue jeans, pointing at a chalkboard. The chalkboard contains several hand-drawn diagrams and text. At the top, 'hemispheric fn. $h(n)$ ' is written. Below it, 'GLOBAL SEARCH' is written. To the left, 'Time Space' is circled, with an arrow pointing to 'depends upon $h(n)$ '. Below this, 'Quality × Completeness' is written. To the right, 'PRIORITY QUEUE' is written, with an arrow pointing to a circle labeled 'OPEN', which has an arrow pointing to it from the right. Below 'OPEN', 'GLOBAL' is written. In the center, there is a search tree diagram with a root node and several levels of children. A thick white line traces a path through the tree. Below the tree, there are two stick figures standing on a wavy line. At the bottom, 'STEEPEST GRADIENT ASCENT' is written. On the right side, 'HILL CLIMBING' is written, followed by 'While []' and 'Next' with a curved arrow.

Heuristic Search — Best First Search

heuristic fn $h(n)$

GLOBAL SEARCH

Time
Space

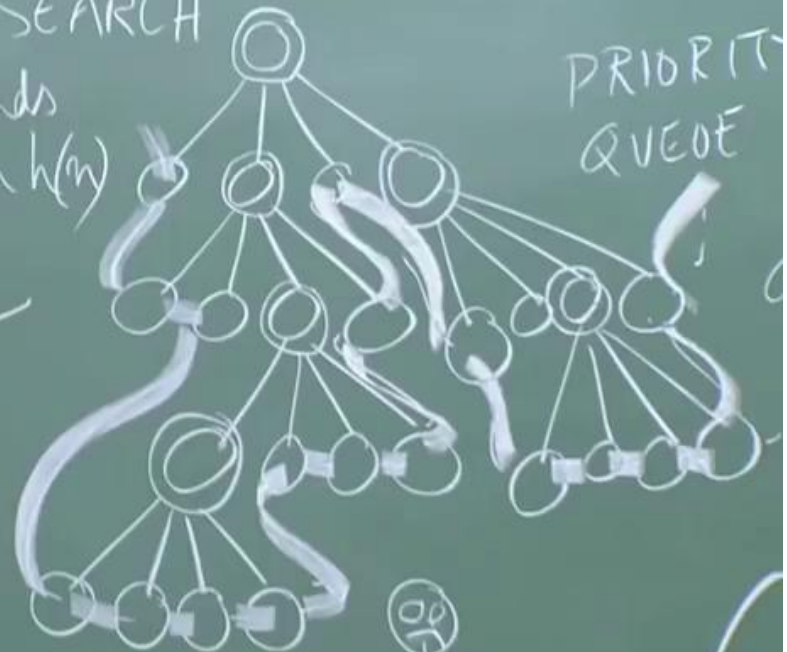
depends
upon $h(n)$

Exponential

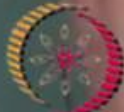
Quality

Completeness

PRIORITY
QUEUE



STEEPEST GRADIENT



NPTEL

instr fn $h(n)$

nodePair \leftarrow Head(OPEN)

OPEN empty

PRIOR
QUEUE

nodePair \leftarrow sort_h(append New Tail(OPEN))



SPACE \rightarrow CONSTANT

CLIMBING

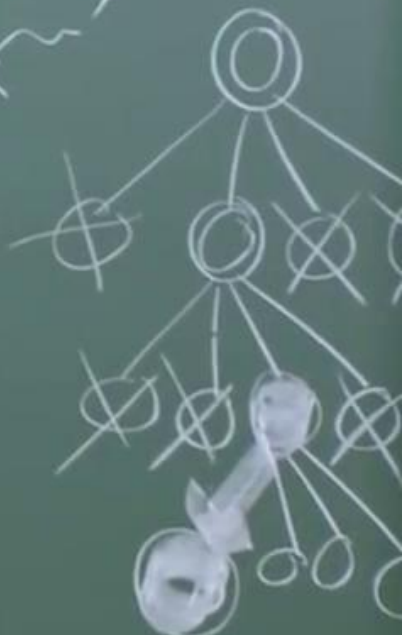
while Next is better than Current

Next \leftarrow Best(moveGen(Current))



OPTIMIZATION - Optimize $h(n)$

ASCENT



heuristic fn $h(n)$

SEARCH

depends upon $h(n)$

PRIORITY QUEUE \leftarrow OPEN \leftarrow sort _{h} (a)

GLOBAL

HILL CLIMBING

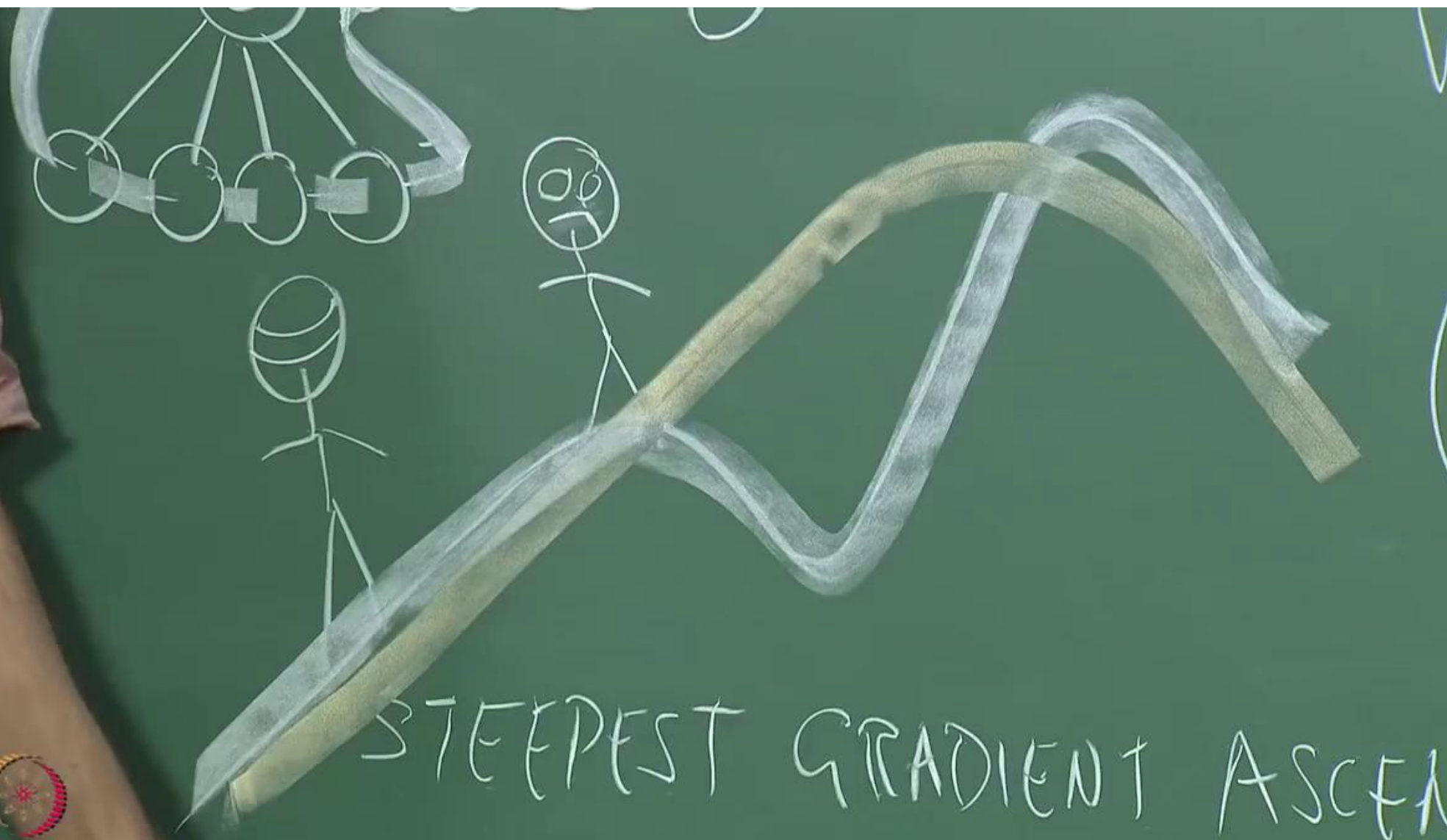
While Next is better

Next \leftarrow Best

OPTIMIZATION

STEEPEST GRADIENT ASCENT

The chalkboard contains several handwritten diagrams and text. At the top left, there is a small tree diagram with nodes labeled 'G'. Below it, the text 'represented' is written. In the center, a large search tree is drawn with a root node and several levels of children. A path is highlighted with a thick white line. To the right of the tree, the text 'PRIORITY QUEUE' is written, followed by an arrow pointing to a circle labeled 'OPEN'. To the right of 'OPEN' is an arrow pointing to the text 'sort _{h} (a)'. Below 'OPEN' is the word 'GLOBAL'. To the right of the tree, the text 'HILL CLIMBING' is written, followed by a 'While' loop containing 'Next is better'. Below the loop, the text 'Next -> Best' is written, followed by an arrow pointing down to the word 'OPTIMIZATION'. At the bottom, a graph is drawn with a wavy line representing a hill. Two stick figures are shown on the line, one at a local minimum and one at a local maximum. Below the graph is the text 'STEEPEST GRADIENT ASCENT'. The man is standing on the left side of the chalkboard, looking at the camera with his hand on his chin.



STEEPEST GRADIENT ASCENT

BLOCKS WORLD DOMAIN





BLOCKS WORLD DOMAIN - $\text{Move}(X, S, D)$



BLOCKS WORLD DOMAIN - $\text{Move}(X, S, D)$





⑤

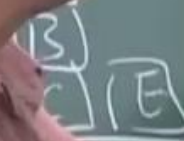
insertion fn. $h_1(n)$

$h_1(n)$ = add 1 if block is on correct block / table
subtract 1 if it is on a wrong location



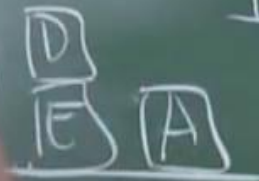
NPTEL

BLOCKS WORLD DOMAIN - $\text{Move}(X, S, D)$

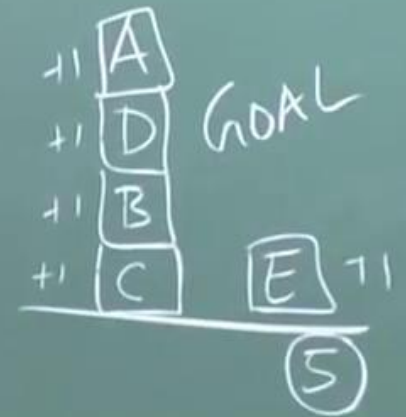
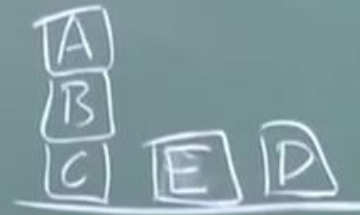


BLOCKS WORLD DOMAIN - $\text{Move}(X, S, D)$

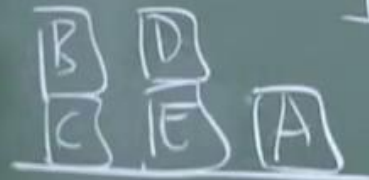
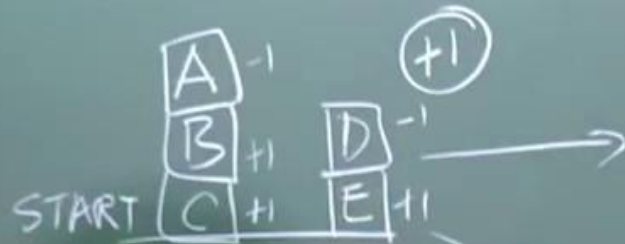
START



BLOCKS WORLD DOMAIN - Move(X,S,D)



BLOCKS WORLD DOMAIN - Move(X,S,D)



BLOCKS WORLD DOMAIN - $\text{Move}(X, S, D)$



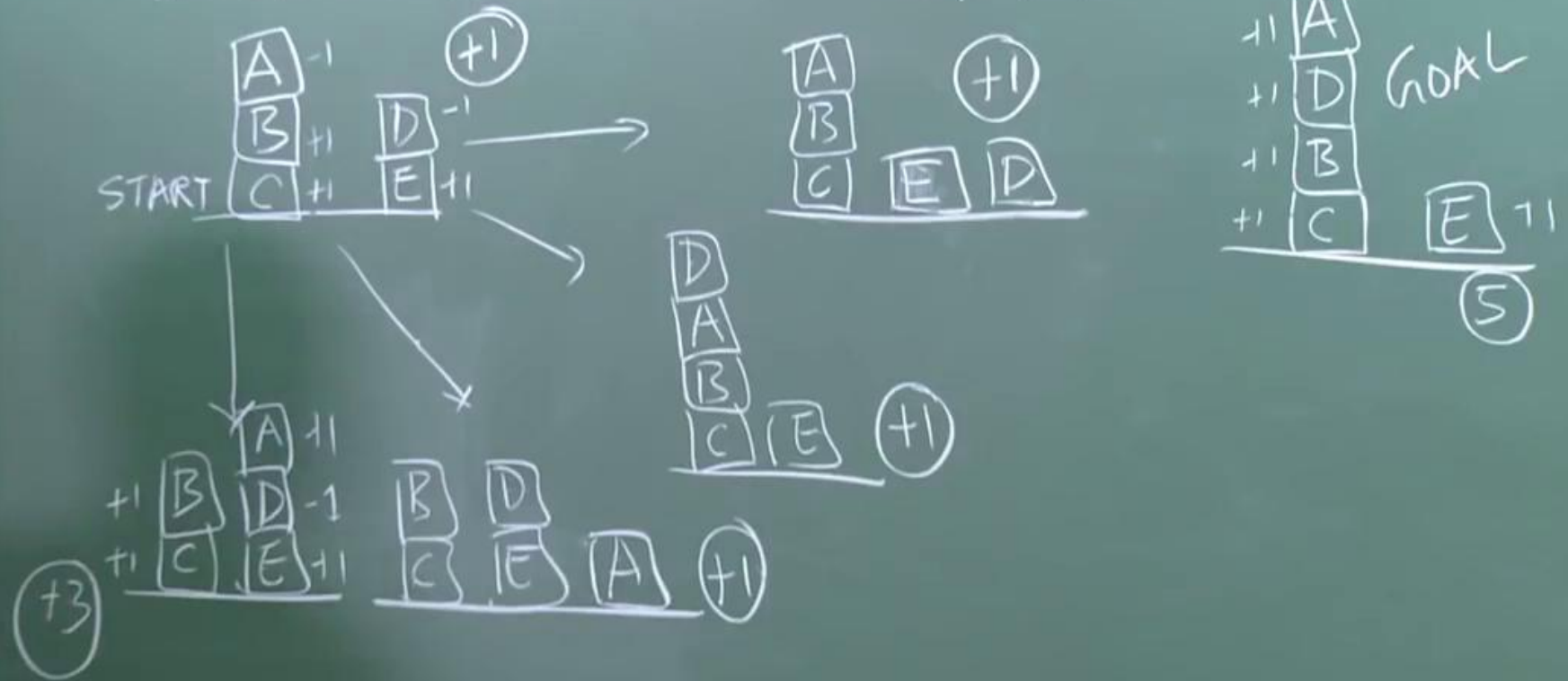
BLOCKS WORLD DOMAIN - Move(X, S, D)



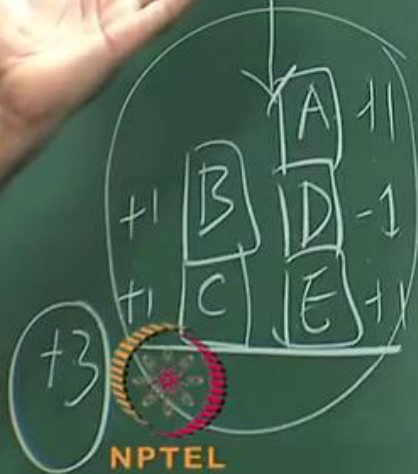
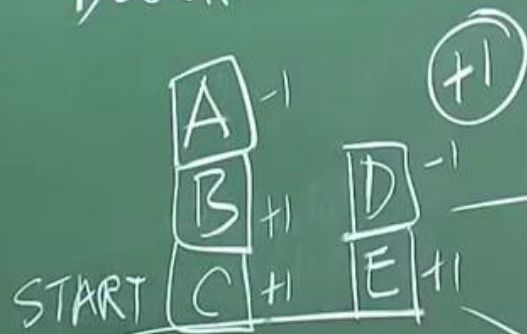
NPTEL

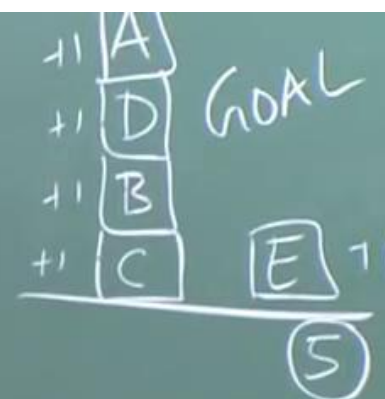
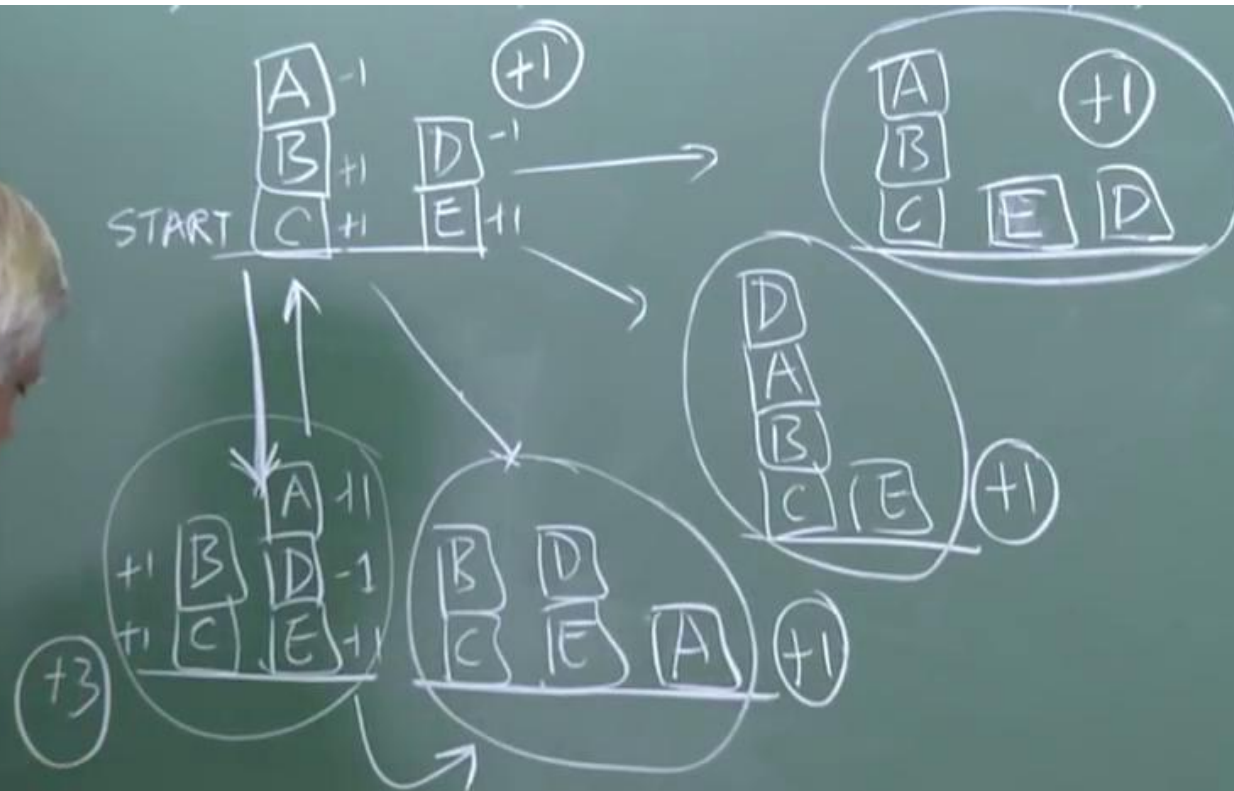


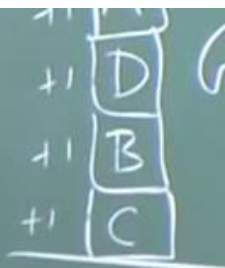
BLOCKS WORLD DOMAIN - Move(X,S,D)



BLOCKS WORLD DOMAIN - Move(X, S, D)







GOAL

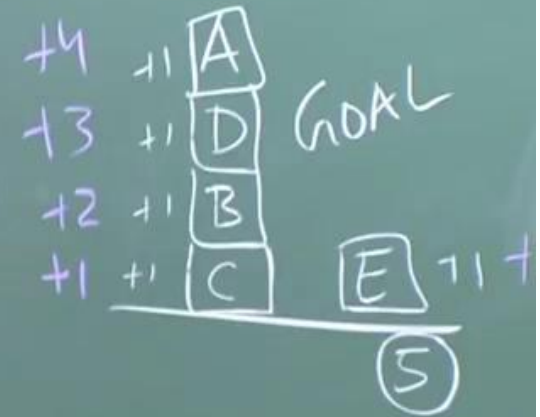
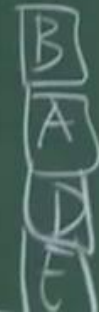
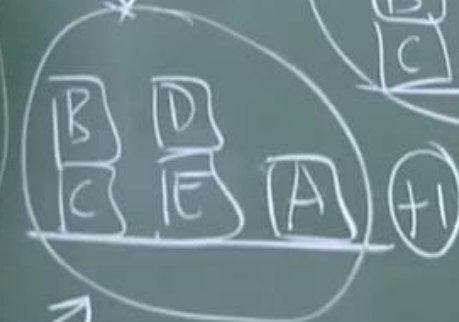
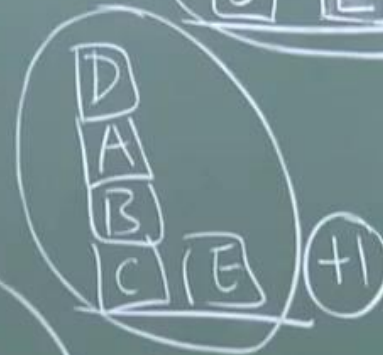
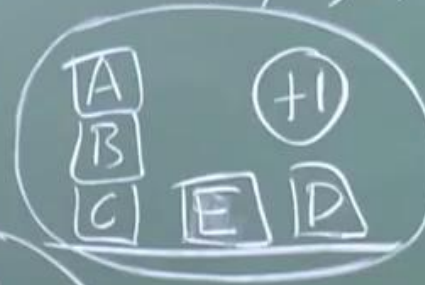
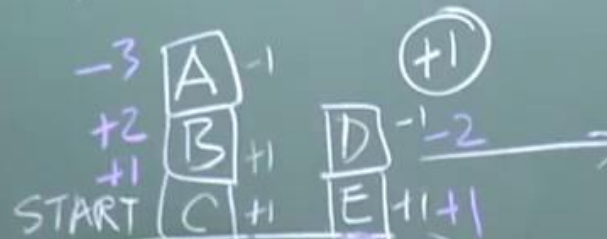
(5)

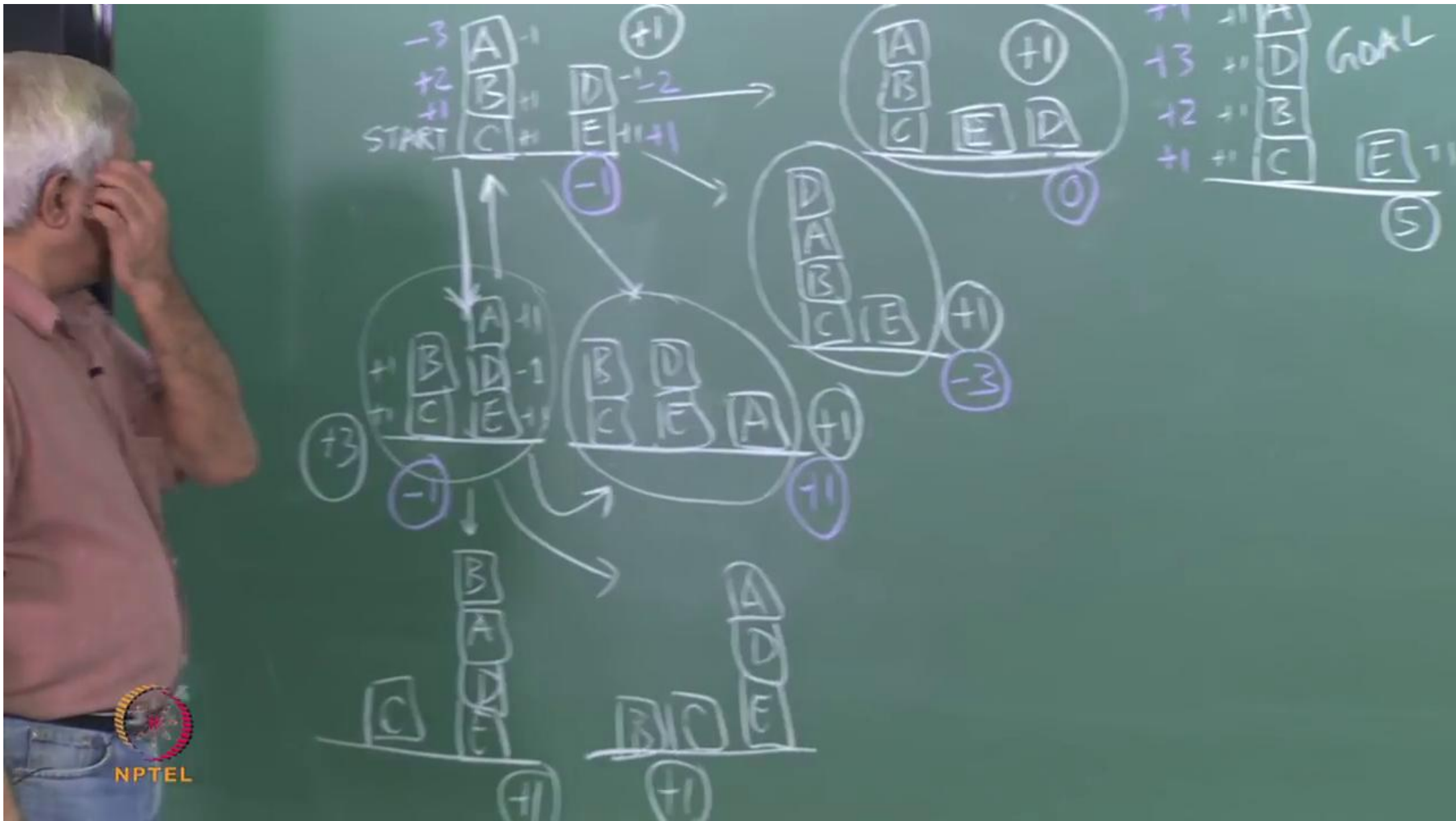
instr. fn. $h(n)$

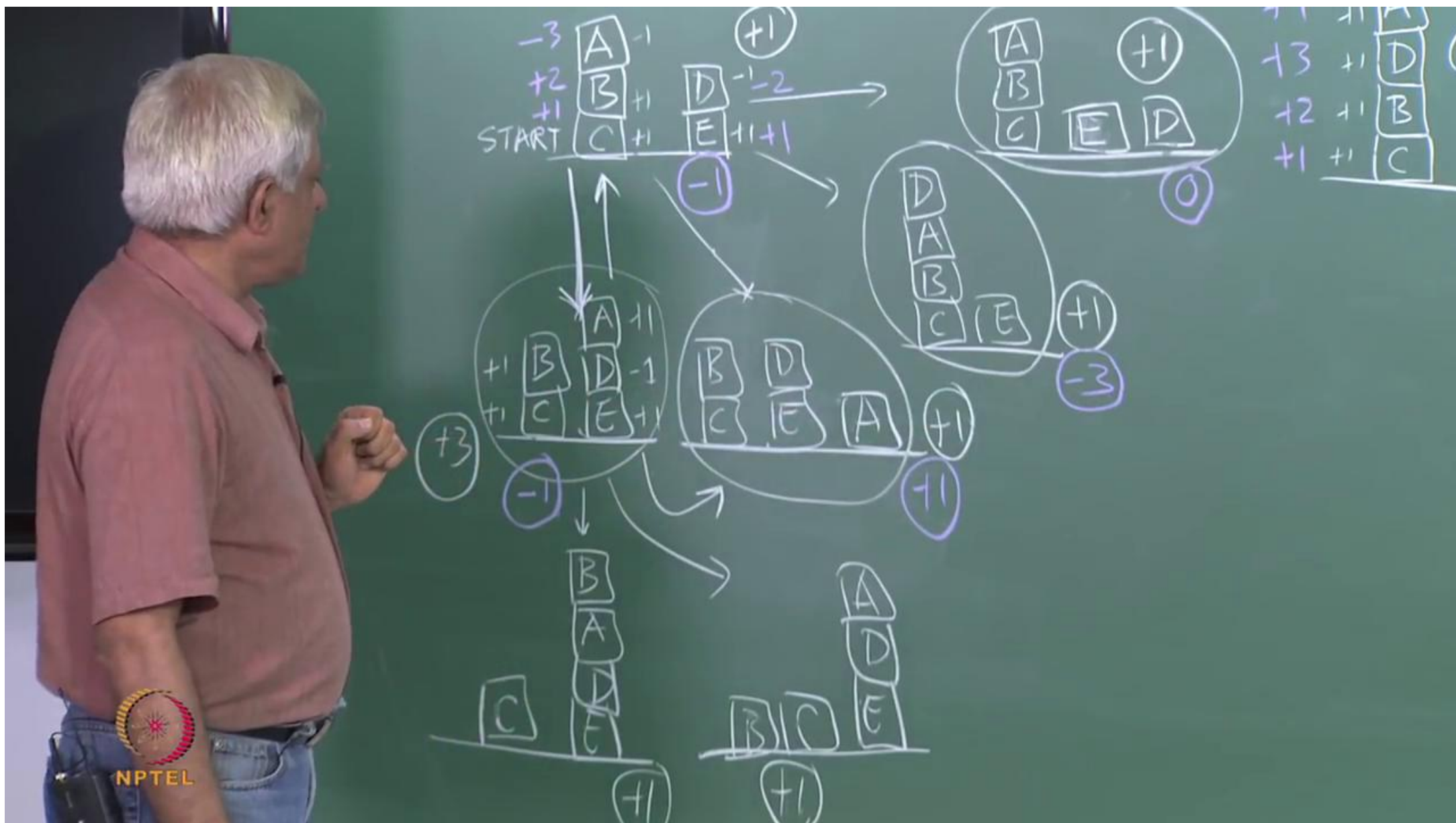
$h_1(n)$ = add 1 if block is on correct block/table
subtract 1 if it is on a wrong location

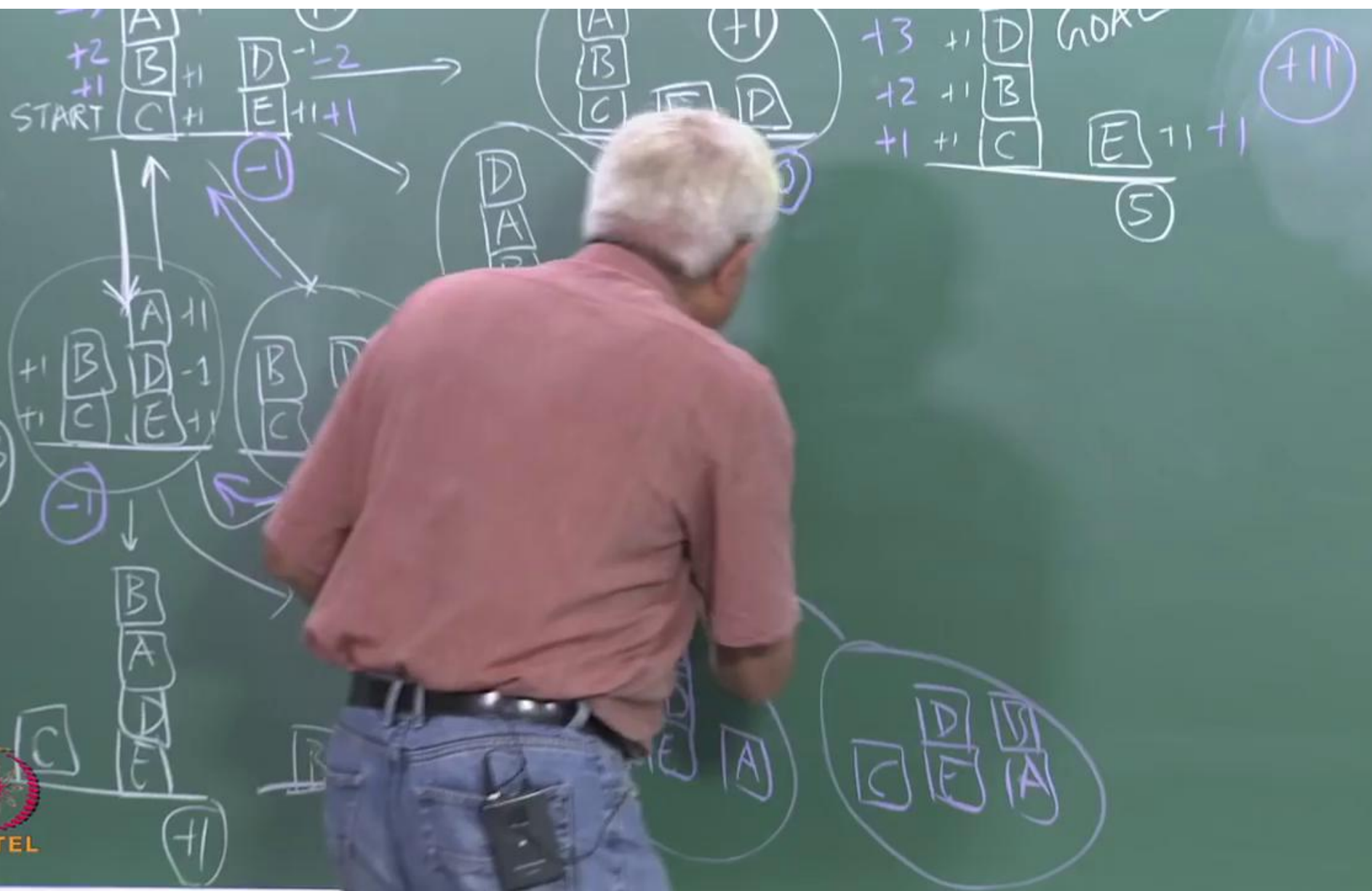
$h_2(n)$ = add 1 for every block in a connection.
STRUCTURE that the block is
sitting on
subtract 1 for every block

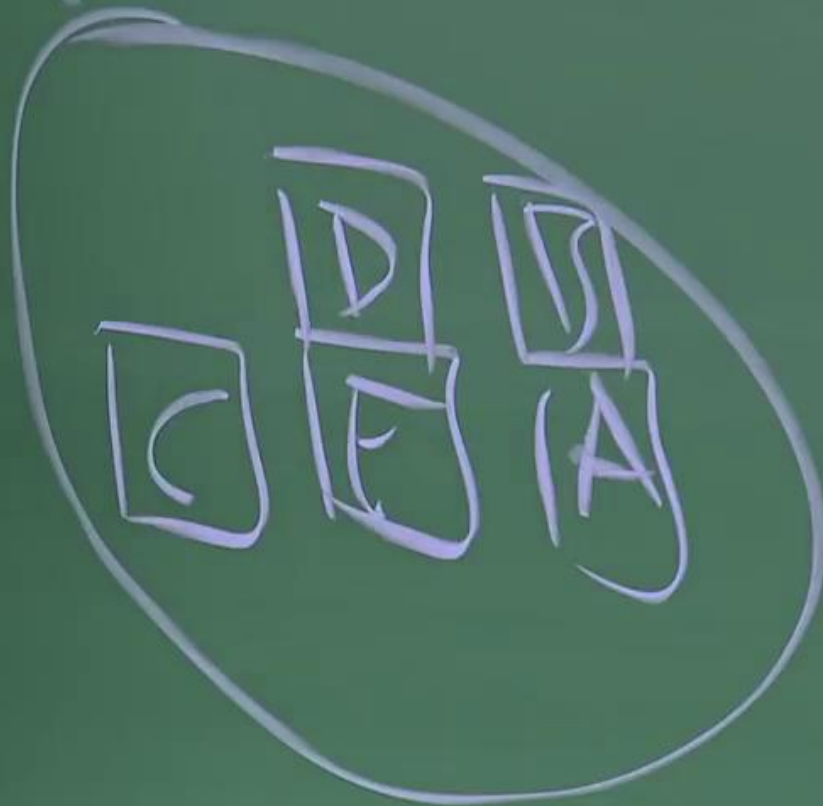
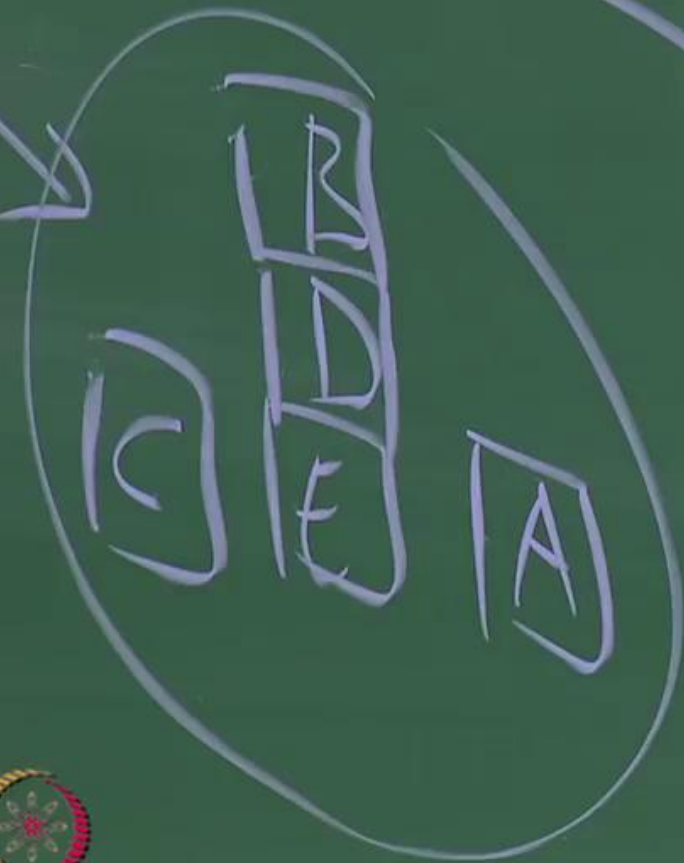
BLOCKS WORLD DOMAIN - Move(X,S,D)

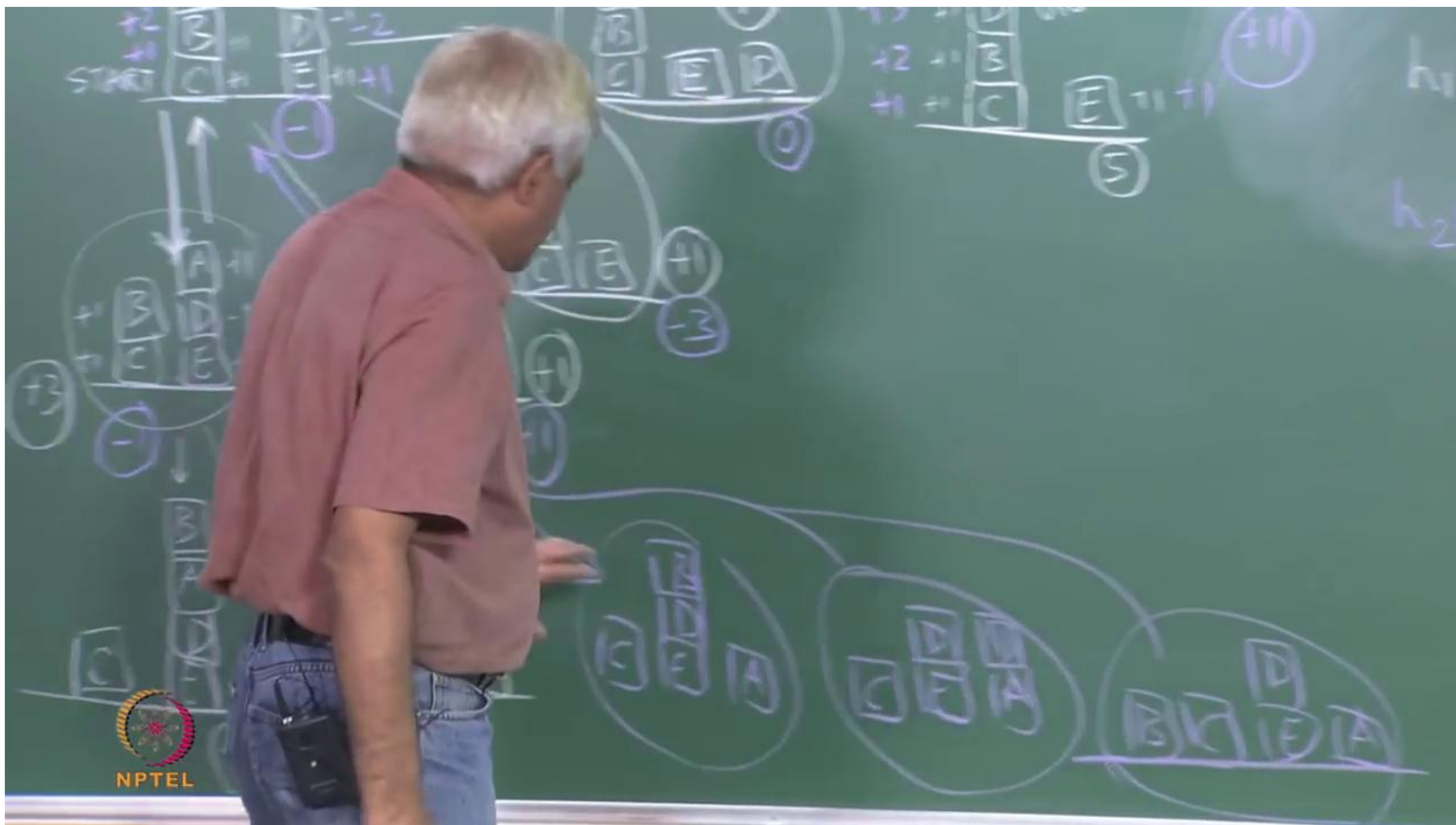


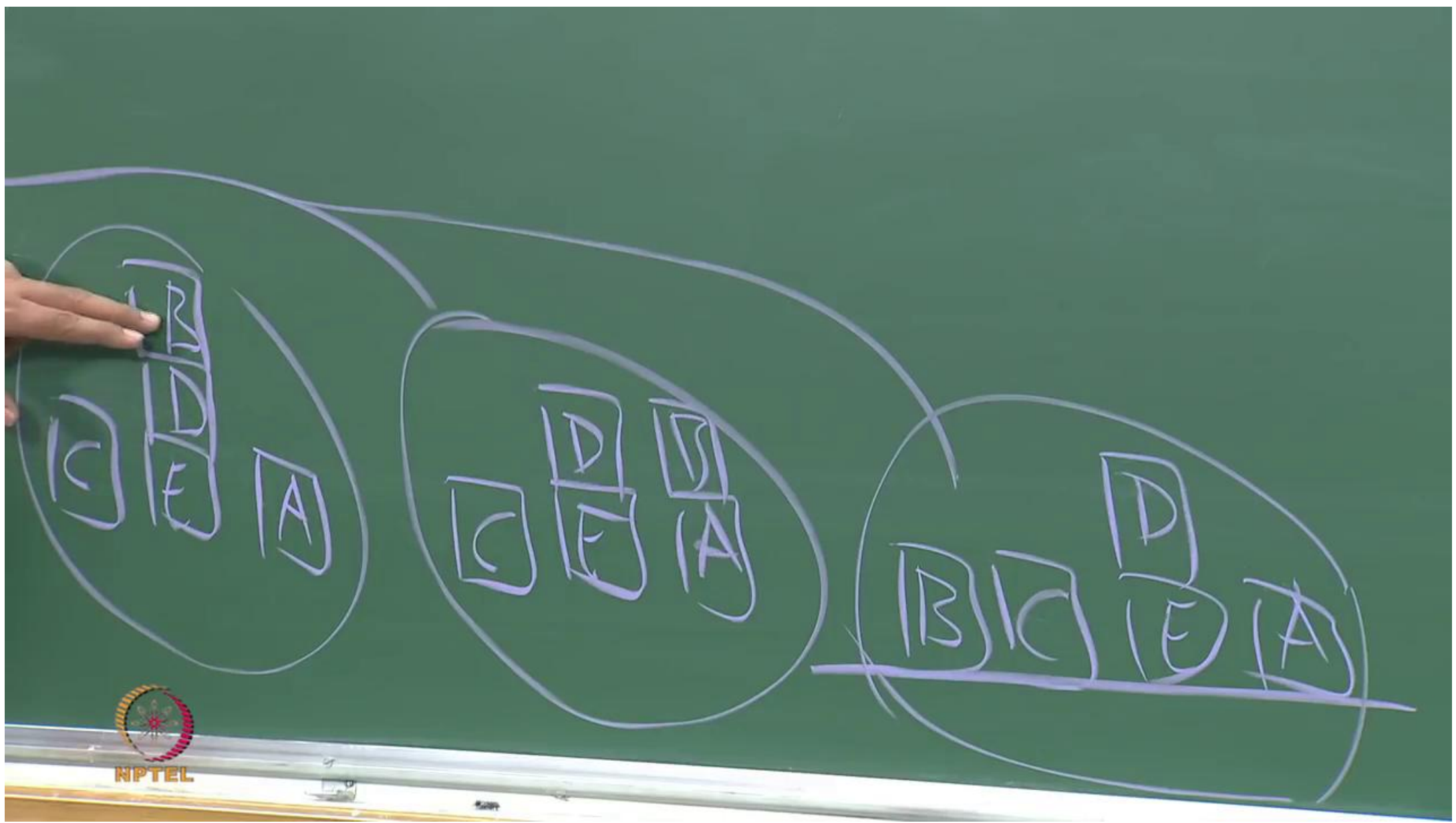












A chalkboard lecture illustrating a sequence of operations on a stack and a queue, likely related to Huffman tree construction or a similar algorithm. The instructor, seen from behind, is pointing at the board.

Stacks and Queues:

- Stacks:** Represented as vertical columns of boxes.
 - Initial stack: $\begin{matrix} D \\ E \end{matrix}$ with values $-1, -2$ and $+1, +1$ next to them.
 - Stack after operation: $\begin{matrix} D \\ A \end{matrix}$ with a value of 0 next to it.
 - Stack after operation: $\begin{matrix} D \\ B \\ C \end{matrix}$ with values $+12, +1$ and $+1, +1$ next to them.
 - Stack after operation: $\begin{matrix} D \\ B \\ C \end{matrix}$ with a value of 5 next to it.
 - Stack after operation: $\begin{matrix} D \\ B \\ C \end{matrix}$ with a value of -3 next to it.
 - Stack after operation: $\begin{matrix} D \\ B \\ C \end{matrix}$ with a value of -2 next to it.
 - Stack after operation: $\begin{matrix} B \\ C \end{matrix}$ with a value of -2 next to it.
- Queues:** Represented as horizontal rows of boxes.
 - Queue after operation: $\begin{matrix} E \\ A \end{matrix}$ with a value of $+11$ next to it.
 - Queue after operation: $\begin{matrix} E \\ A \end{matrix}$ with a value of $+11$ next to it.
 - Queue after operation: $\begin{matrix} E \\ A \end{matrix}$ with a value of $+11$ next to it.
 - Queue after operation: $\begin{matrix} E \\ A \end{matrix}$ with a value of $+11$ next to it.
 - Queue after operation: $\begin{matrix} E \\ A \end{matrix}$ with a value of $+11$ next to it.
 - Queue after operation: $\begin{matrix} E \\ A \end{matrix}$ with a value of $+11$ next to it.
 - Queue after operation: $\begin{matrix} E \\ A \end{matrix}$ with a value of $+11$ next to it.
 - Queue after operation: $\begin{matrix} E \\ A \end{matrix}$ with a value of $+11$ next to it.

Operations:

- $h_1(n) = \text{add}$
- $h_2(n) = \text{add}$
- sub

The NPTEL logo is visible in the bottom left corner.

Diagram 1 (Top Left):

A
E

Goal: +1

Result: 0

Diagram 2 (Top Middle):

D
B
C

Goal: +1

Result: 5

Diagram 3 (Top Right):

Goal: +11

Result: 11

Diagram 4 (Middle):

D
B
C

Result: 3

Diagram 5 (Bottom Left):

D
B
C

Result: -3

Diagram 6 (Bottom Middle):

D
B
C

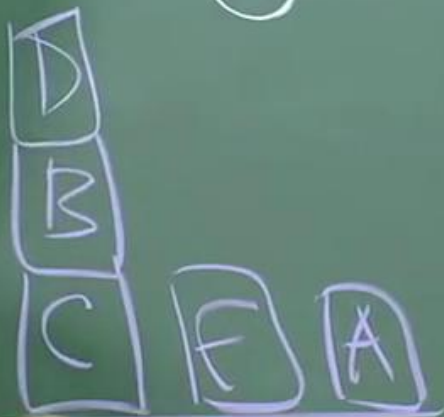
Result: -3

Diagram 7 (Bottom Right):

B
C

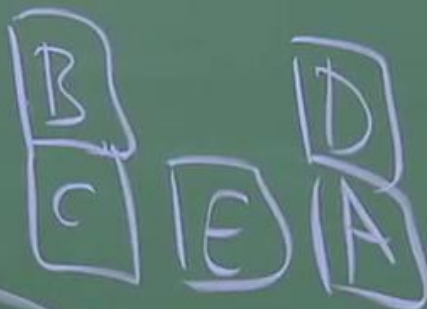
Result: 11

$h_1(n) = \text{add 1 if } b_n = 1$
 $h_2(n) = \text{add 1 if } b_n = 1$
 $h_3(n) = \text{add 1 if } b_n = 1$



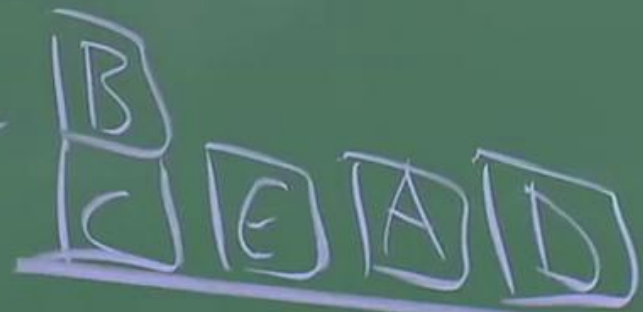
$h_2(n)$ = add 1 for every block
STRUCTURE

subtract 1 for every



(-3)

(-2)



A professor is standing in front of a chalkboard, illustrating a sorting algorithm using blocks labeled A, B, C, D, and E. The board contains several diagrams and handwritten notes.

Diagrams on the left side of the board:

- Top: A stack of blocks A, B, C with a circled $+1$ next to it. Below it, blocks E and D are shown with a circled 0 .
- Middle: A stack of blocks D, A, B, C with a circled $+1$ next to it. Below it, block E is shown with a circled -3 .
- Bottom left: A stack of blocks A, D, E with a circled $+1$ next to it. Below it, a stack of blocks R, D, E is shown with a circled $+1$ next to it.

Handwritten notes on the right side of the board:

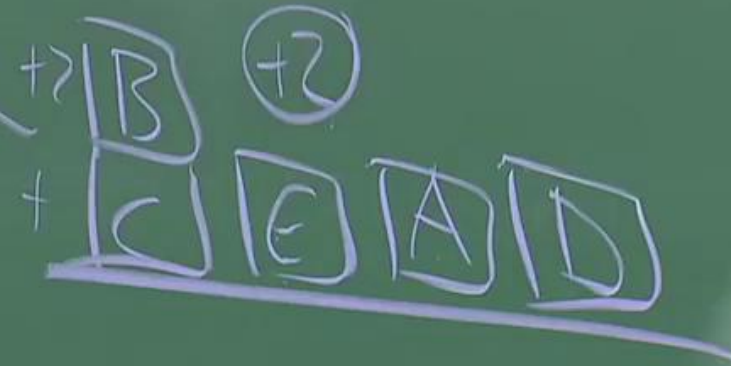
- Top right: $h_1(n)$ = add 1 if block is ... subtract 1 if it is ...
- Middle right: $h_2(n)$ = add 1 for every block STRUCTURE
- Bottom right: A stack of blocks B, C, E, A, D with a circled $+2$ next to it.

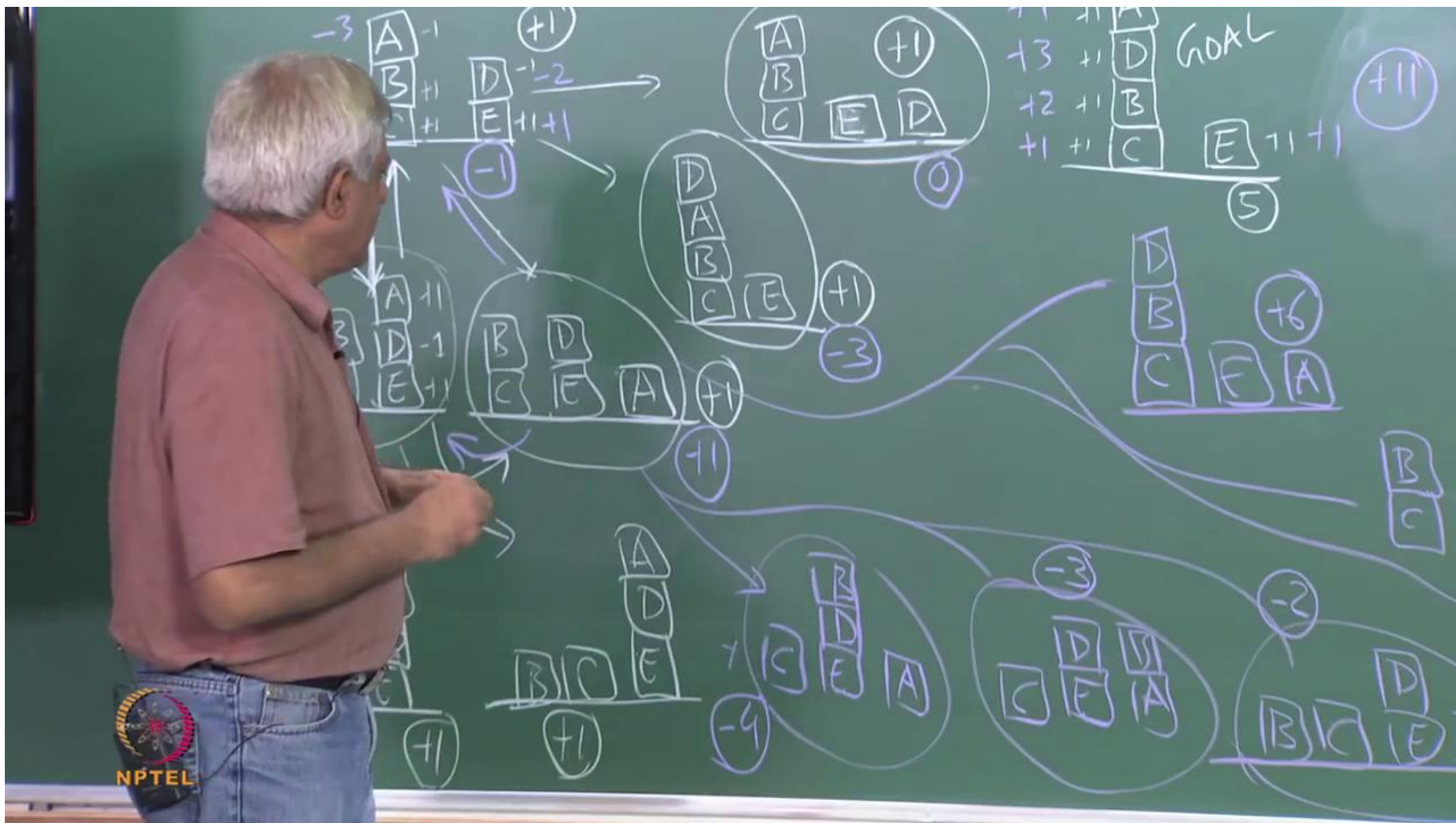
The professor is pointing at a diagram in the center of the board, which shows a stack of blocks B, C, E, A, D.

In the bottom left corner, there is a logo for NPTEL (National Programme on Technology Enhanced Learning).

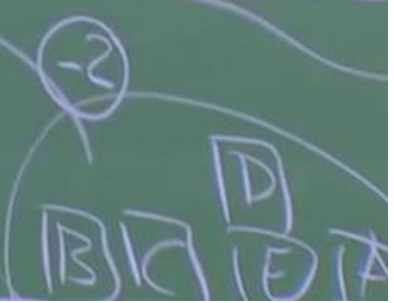
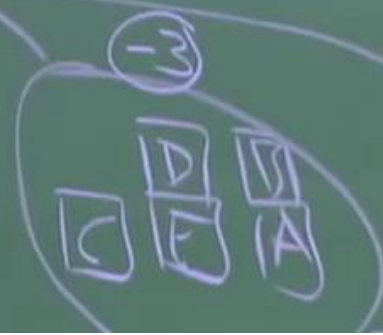
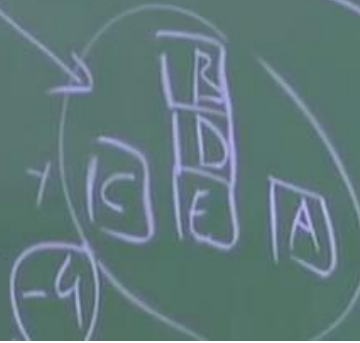
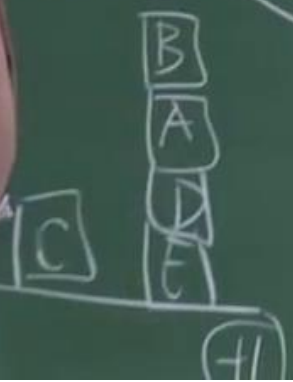
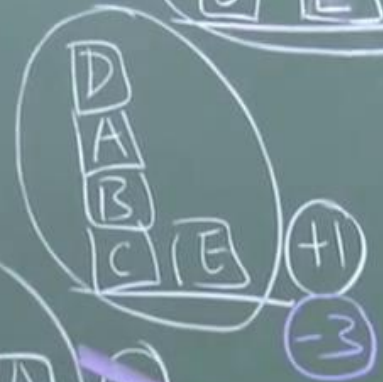
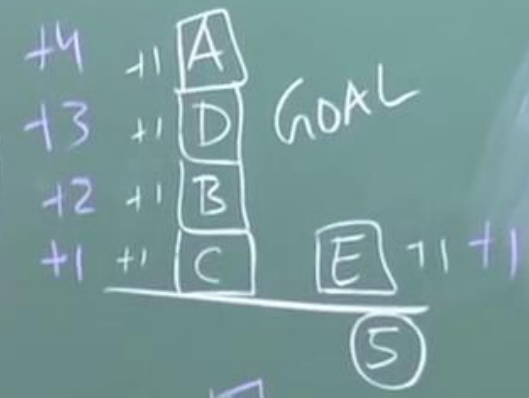
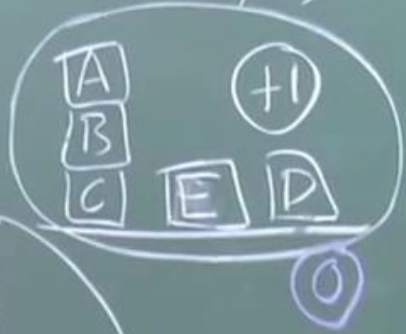
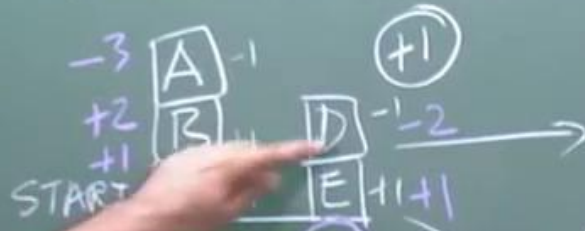
$h_2(n) = \text{add 1 for every block}$
STRUCTURE

subtract 1 for every





BLOCKS WORLD DOMAIN - Move(X,S,D)



BLOCKS WORLD DOMAIN - Move(X, S, D)

