Title: Super-polynomial Speed-up for a Quantum Computer on Boolean Trees

Date: Jul 20, 2011 09:50 AM

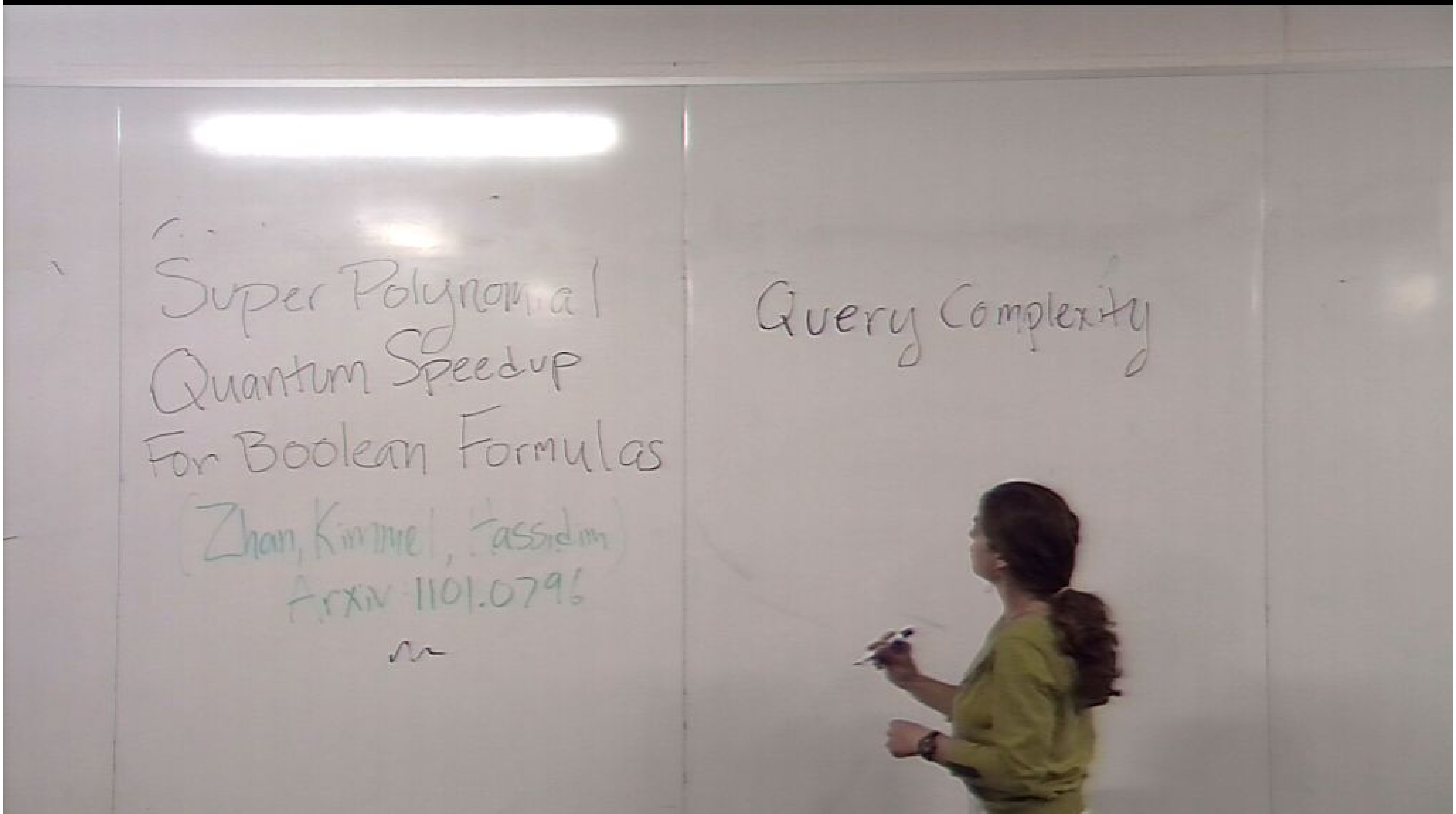
URL: http://pirsa.org/11070063

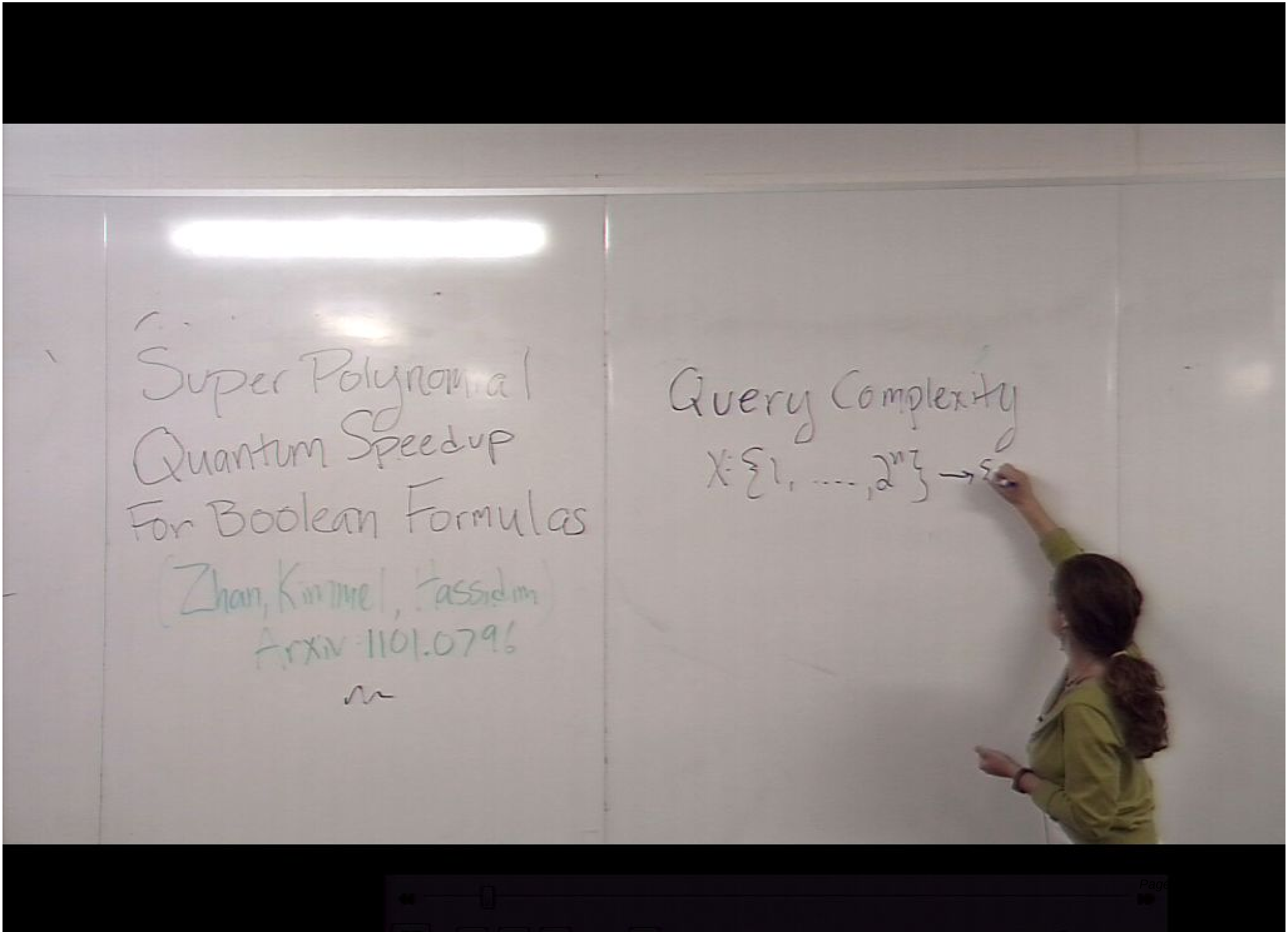
Abstract: We can prove that for certain problems, quantum computers do better than classical computers. I will introduce the query complexity framework, which lets us compare classical and quantum computers, and then describe a problem where quantum computers do better than classical. The problem I will discuss is evaluating boolean trees with a promise on the input.

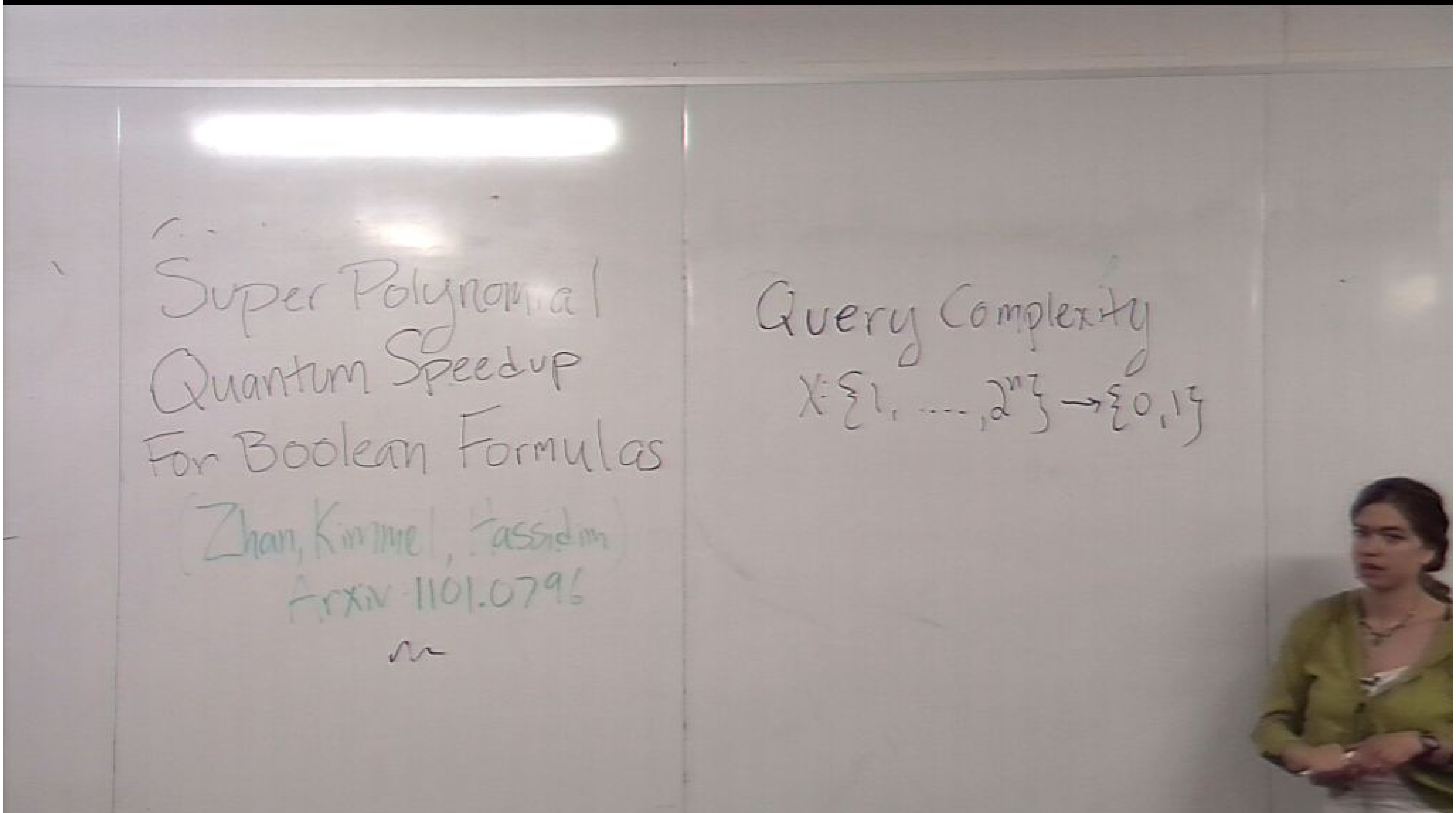
Super Polynomial Quantum Speedup For Boolean Formulas

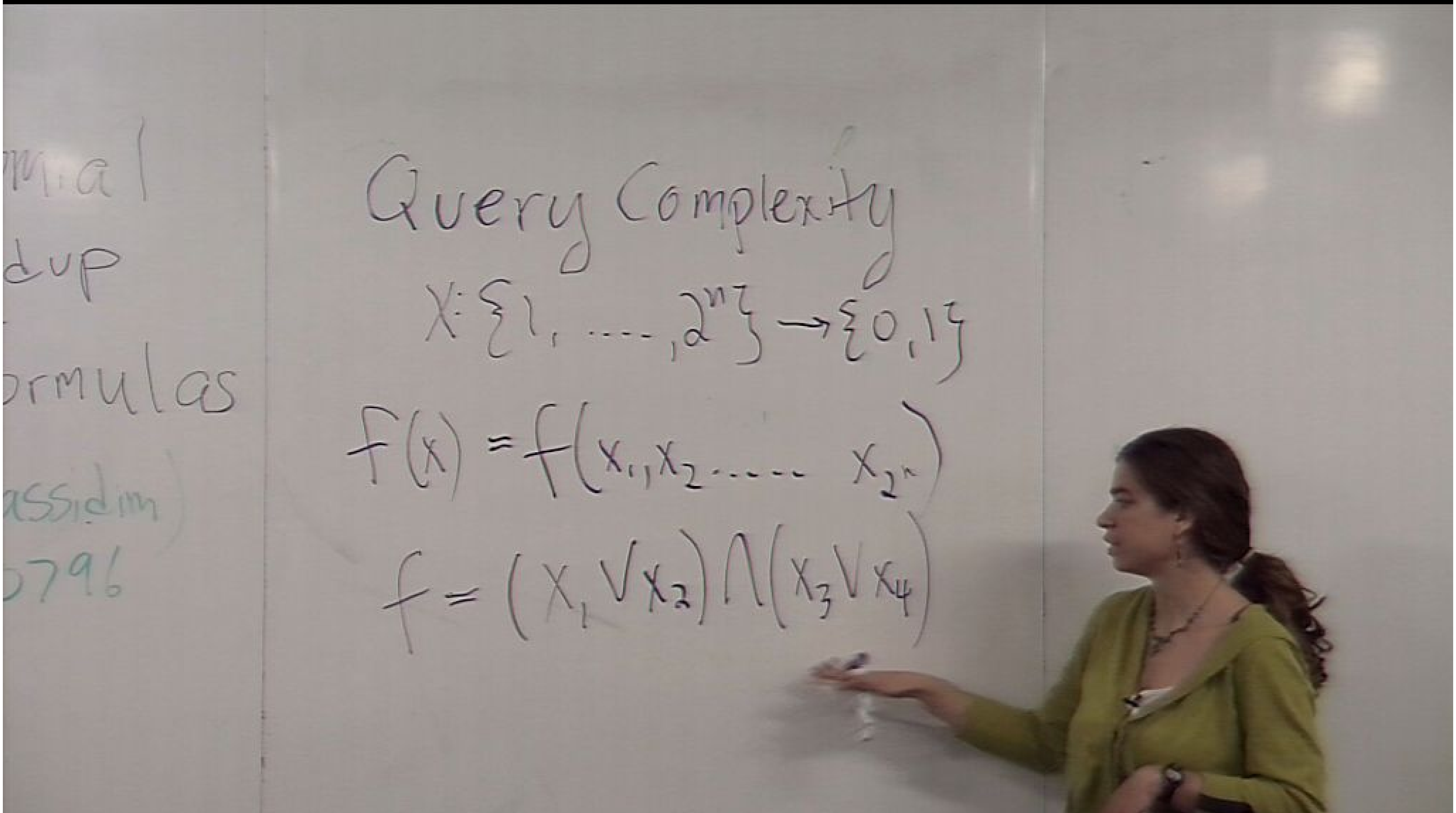
(Zhan, Kimmel, Hassidim)

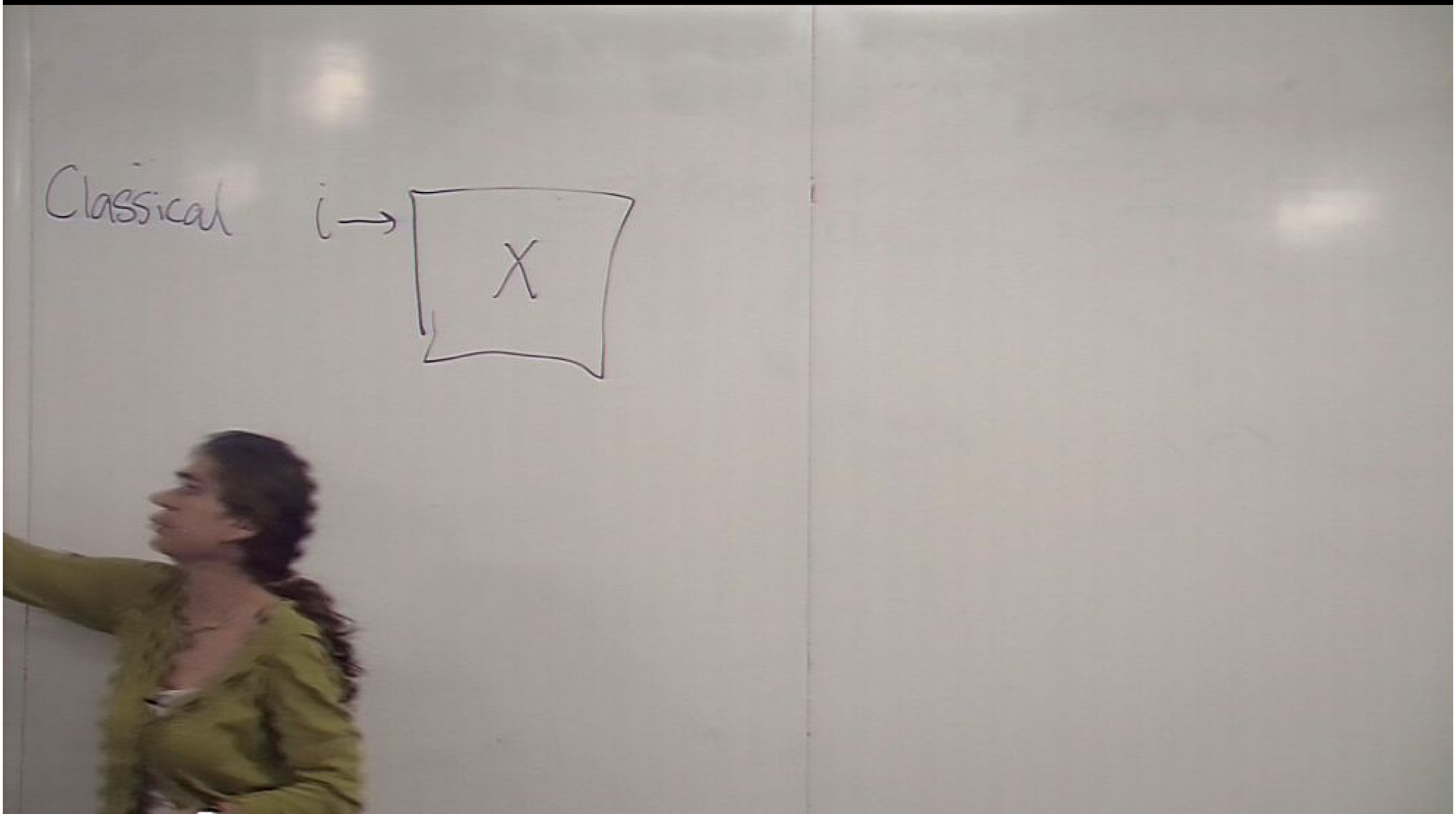
Arxiv 1101.0796

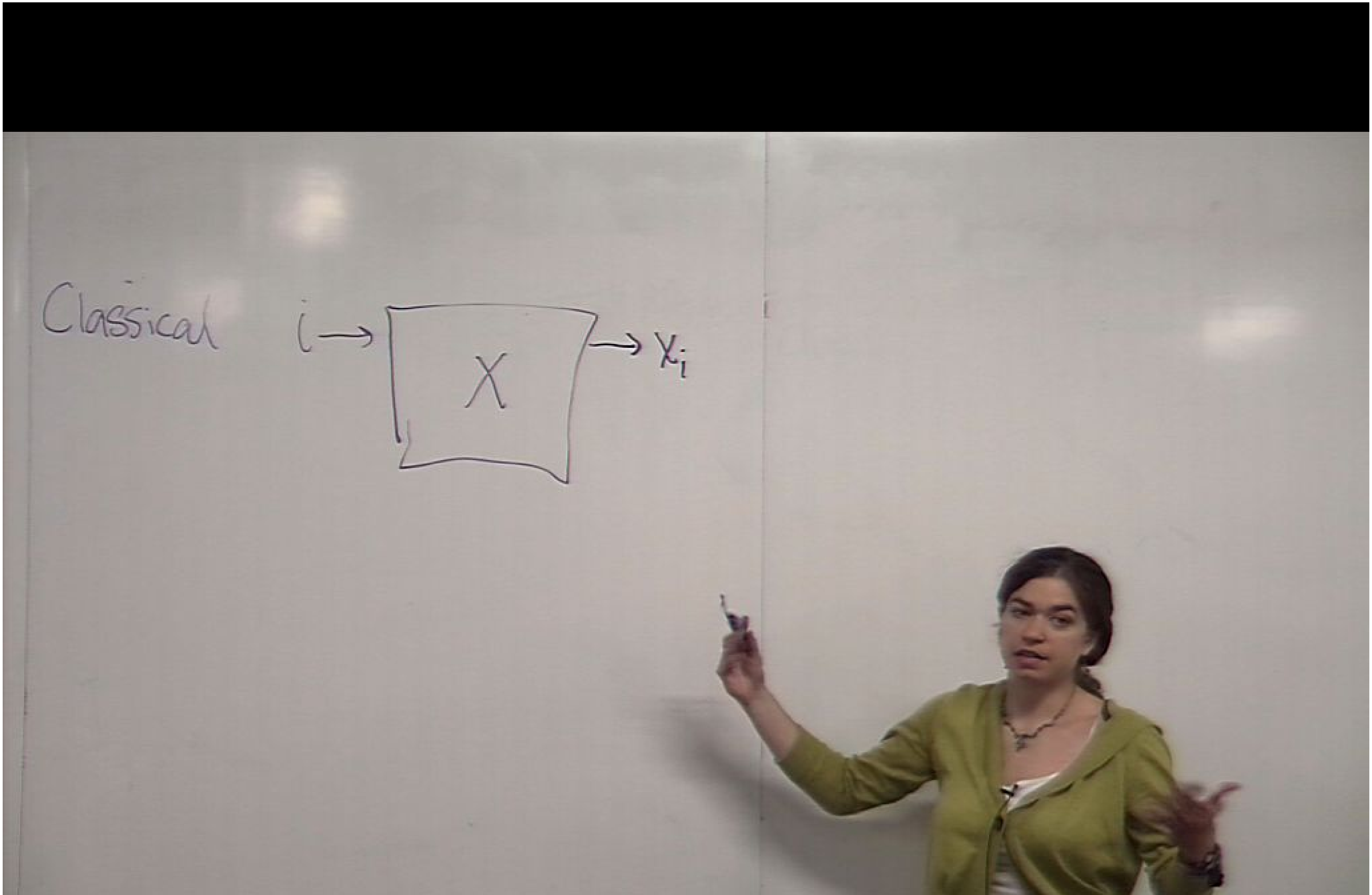

Super Polynomial
Quantum Speedup
For Boolean Formulas
(Zhan, Kimmel, Hassidim)
Arxiv 1101.0796
Query Complexity

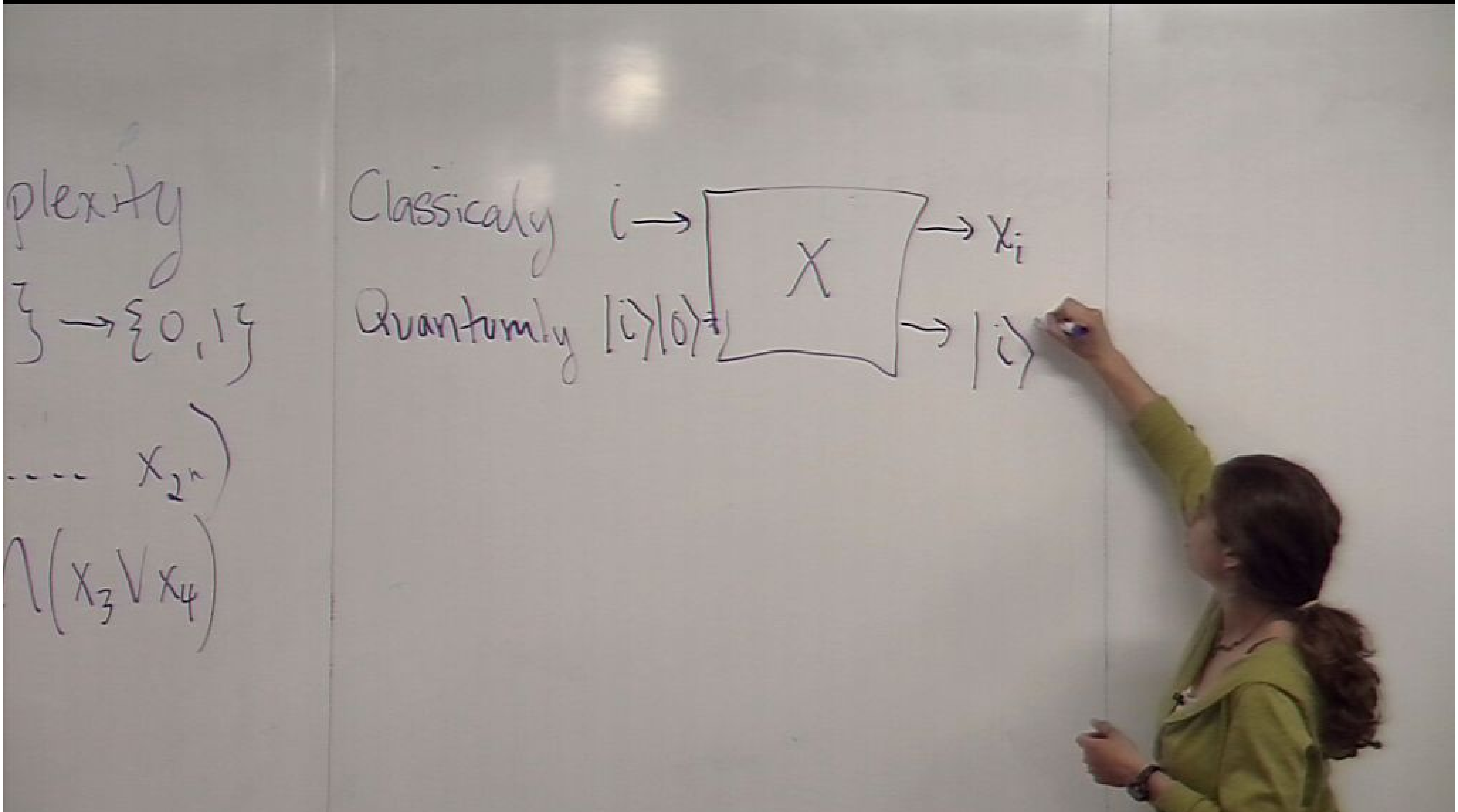

Super Polynomial
Quantum Speedup
For Boolean Formulas
(Zhan, Kimmel, Hassidim)
Arxiv 1101.0796
Query Complexity
X: {1, ..., 2^n} → {

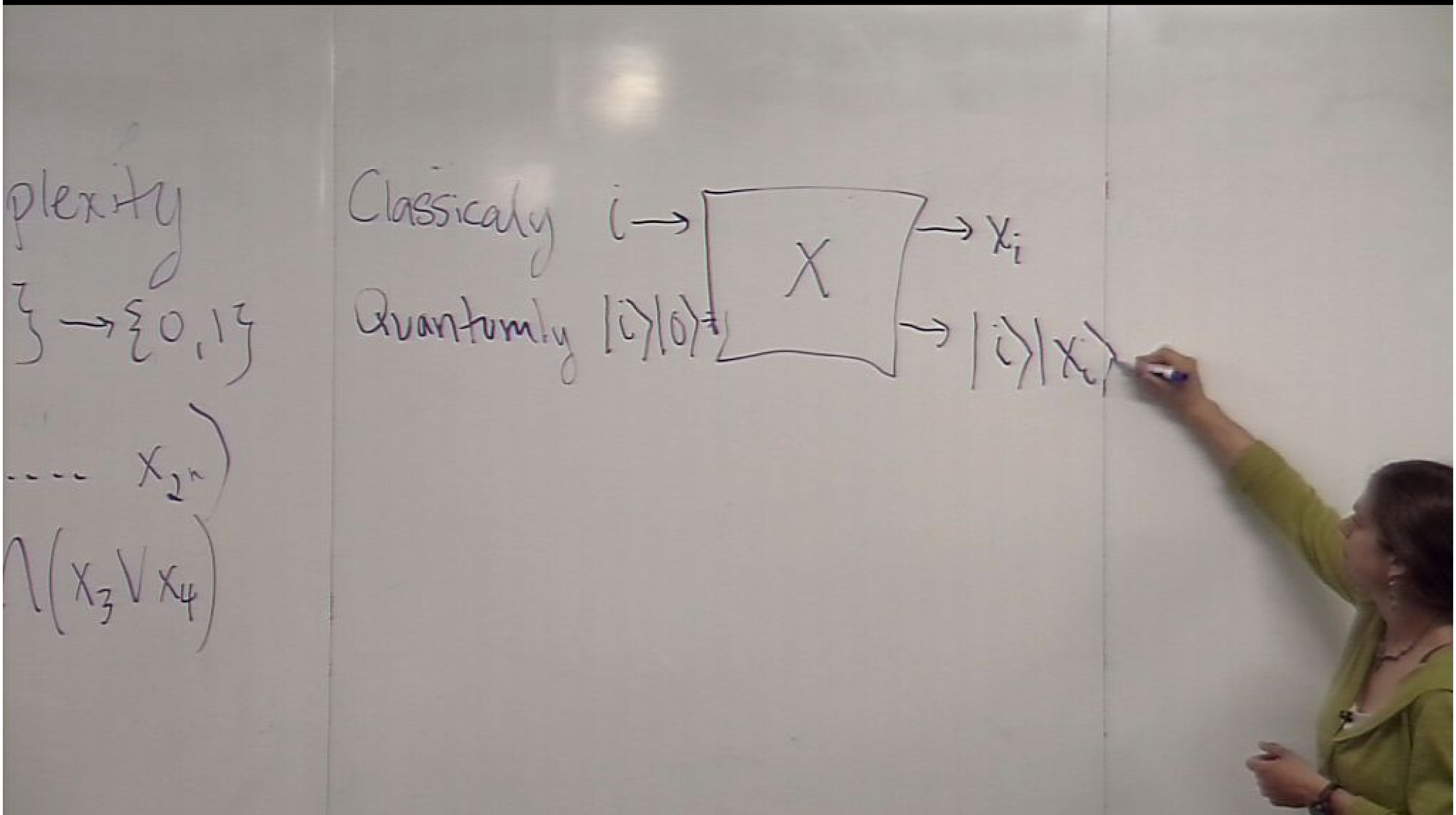

Super Polynomial
Quantum Speedup
For Boolean Formulas
(Zhan, Kimmel, Hassidim)
Arxiv 1101.0796
Query Complexity
$x: \{1, \ldots, 2^n\} \to \{0,1\}$

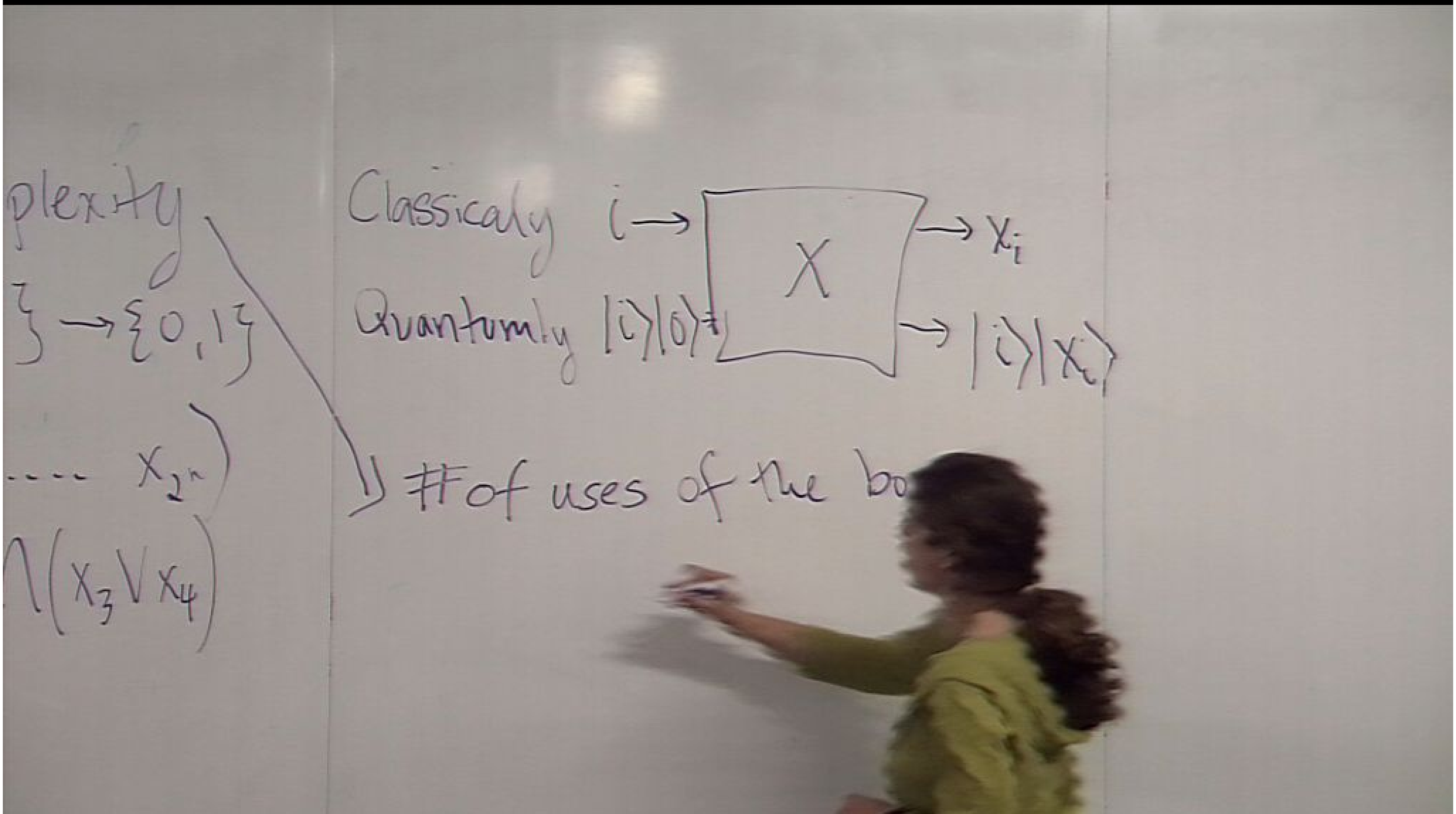

Query Complexity
$X: \{1, \ldots, 2^n\} \rightarrow \{0,1\}$
$f(X) = f(x_1, x_2, \ldots, x_{2^n})$
$f = (x_1 \vee x_2) \wedge (x_3 \vee x_4)$


Classical
i
X


Classical
i →
X
→ X_i


plexity
$\} \to \{0,1\}$
$\ldots\ldots x_{2^n})$
$\wedge(x_3 \vee x_4)$
Classicaly $i \to$
X
$\to x_i$
Quantumly $|i\rangle|0\rangle$
$\to |i\rangle$


plexity
} → {0,1}
.... x_{2^n})
∧(x_3 ∨ x_4)
Classicaly i → X → x_i
Quantumly $|i\rangle|0\rangle$ → X → $|i\rangle|x_i\rangle$


plexity
} → {0,1}
.... x_{2^n})
∧(x_3 ∨ x_4)
Classicaly i → X → x_i
Quantumly |i⟩|0⟩ → |i⟩|x_i⟩
#of uses of the

plexity

$\} \to \{0,1\}$

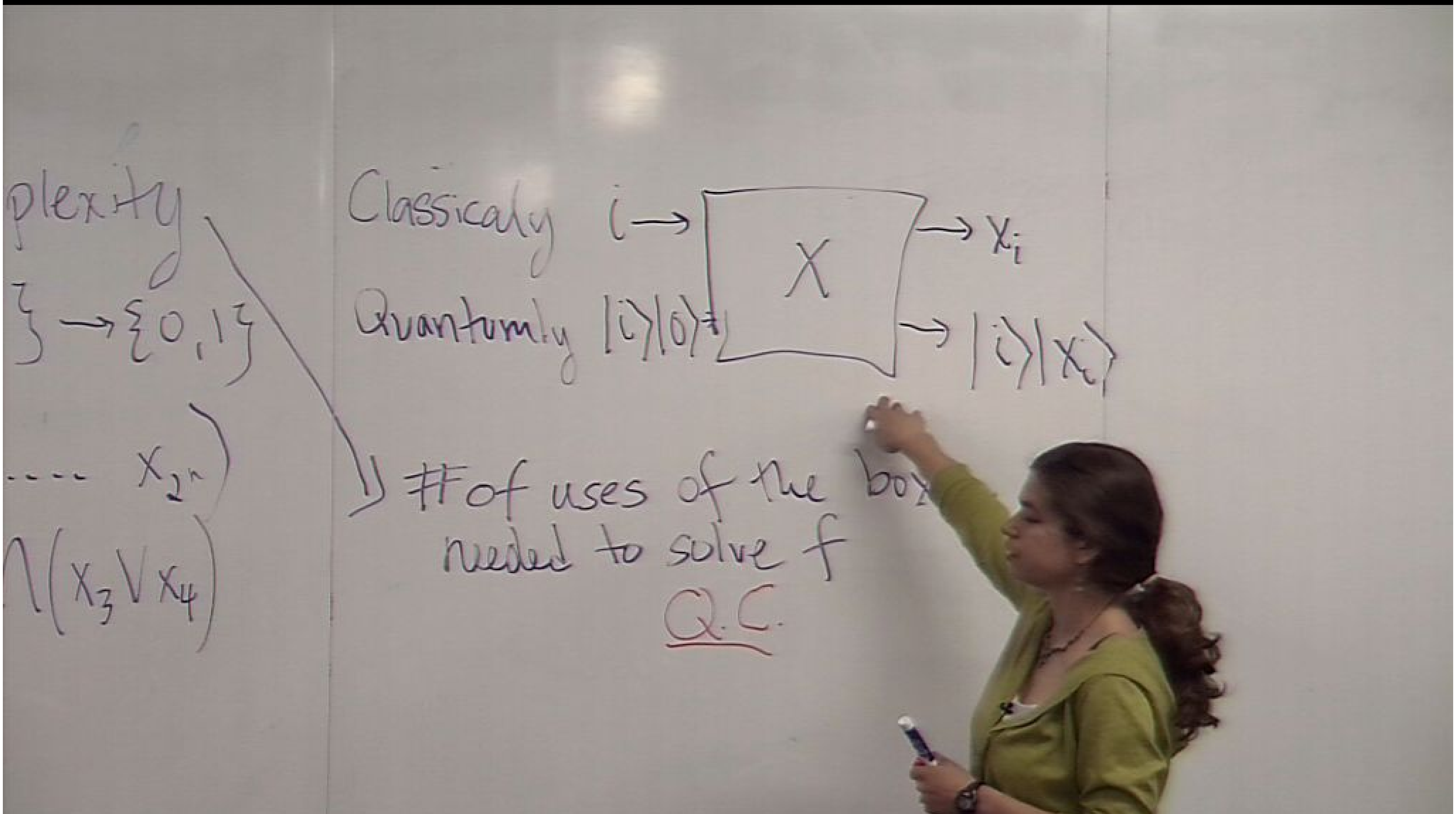
$\ldots\ x_{2^n})$

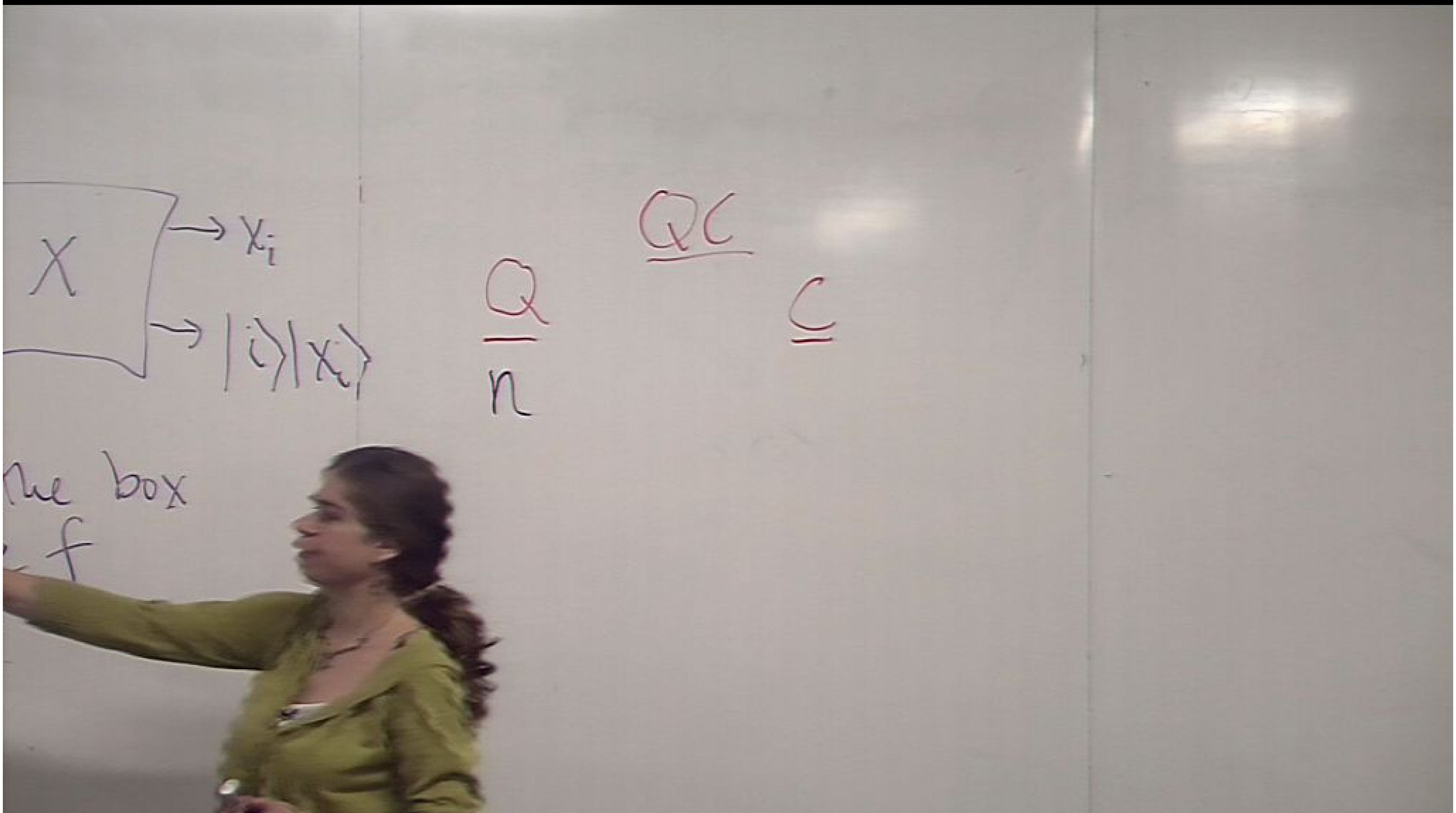
$\wedge(x_3 \vee x_4)$

Classicaly $i \to \boxed{X} \to x_i$

Quantumly $|i\rangle|0\rangle \to \boxed{X} \to |i\rangle|x_i\rangle$

#of uses of the box needed to solve f


plexity
$\} \to \{0,1\}$
$\ldots\ldots\ x_{2^n})$
$\wedge(x_3 \vee x_4)$
Classicaly $i \to$
X
$\to x_i$
Quantumly $|i\rangle|0\rangle \to$
$\to |i\rangle|x_i\rangle$
# of uses of the box
needed to solve f
Q.C.


X
x_i
$|i\rangle|x_i\rangle$
QC
Q
n
C
the box
f

$X \to x_i$

$X \to |i\rangle|x_i\rangle$

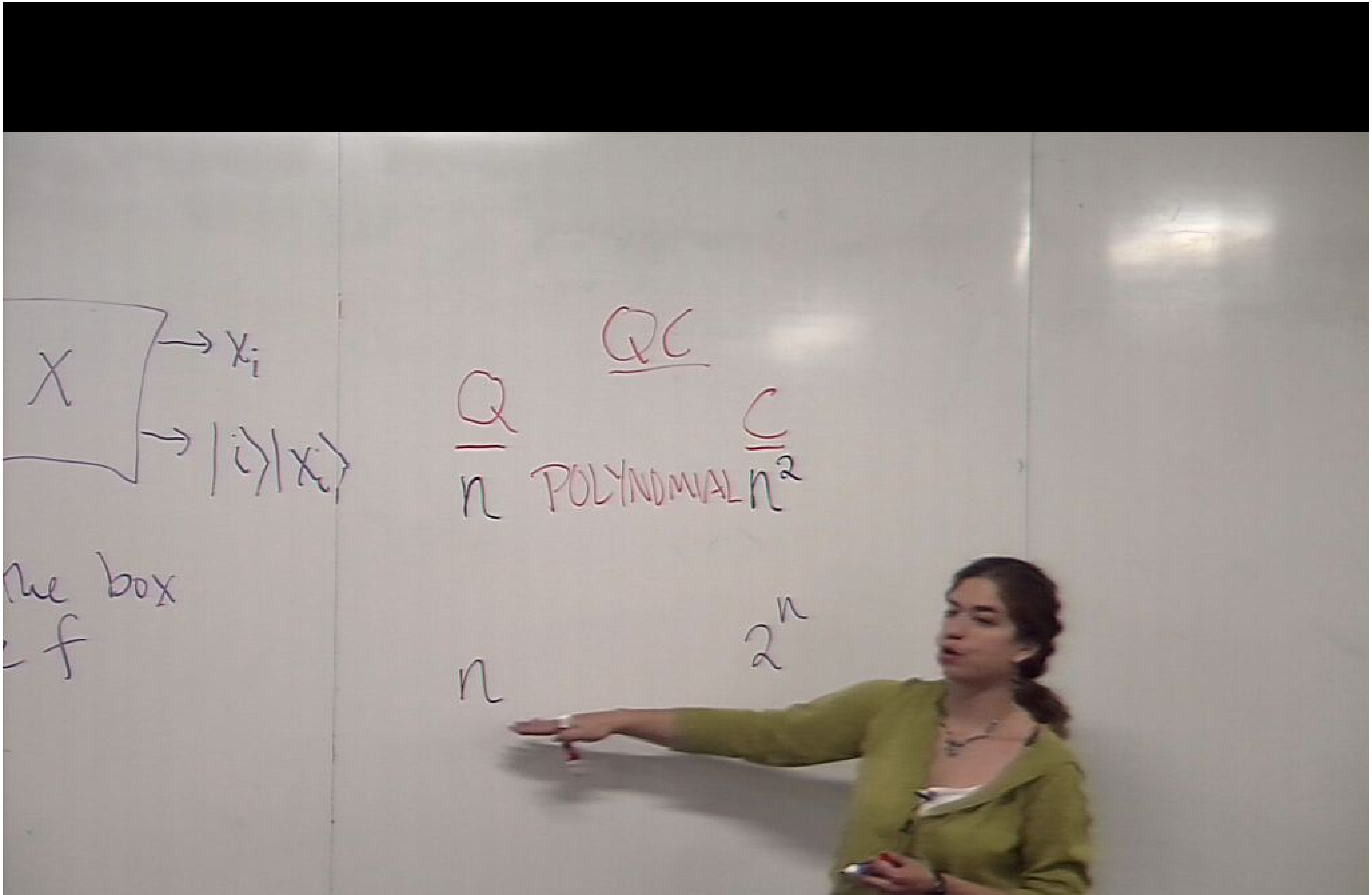
the box

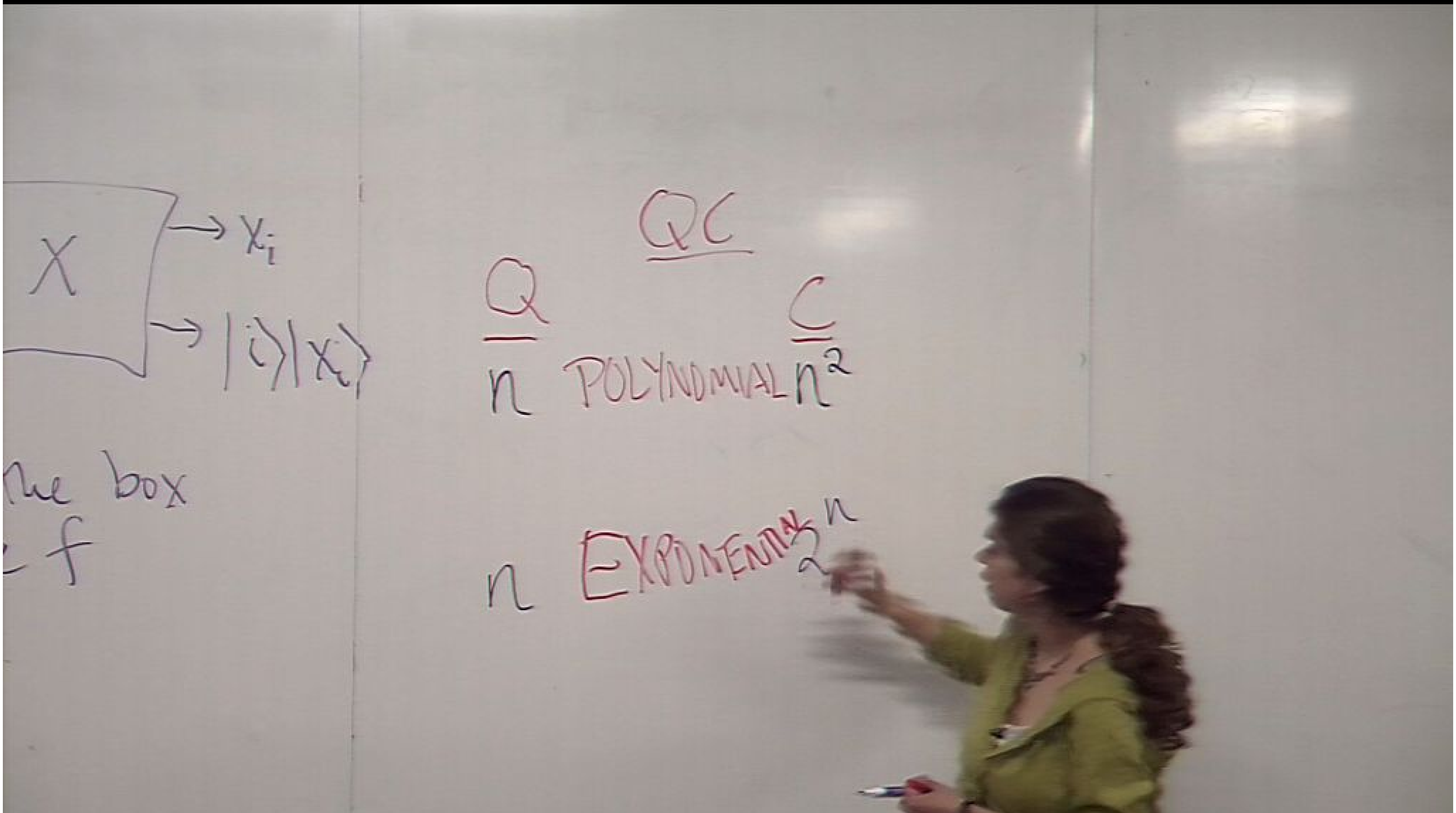
f

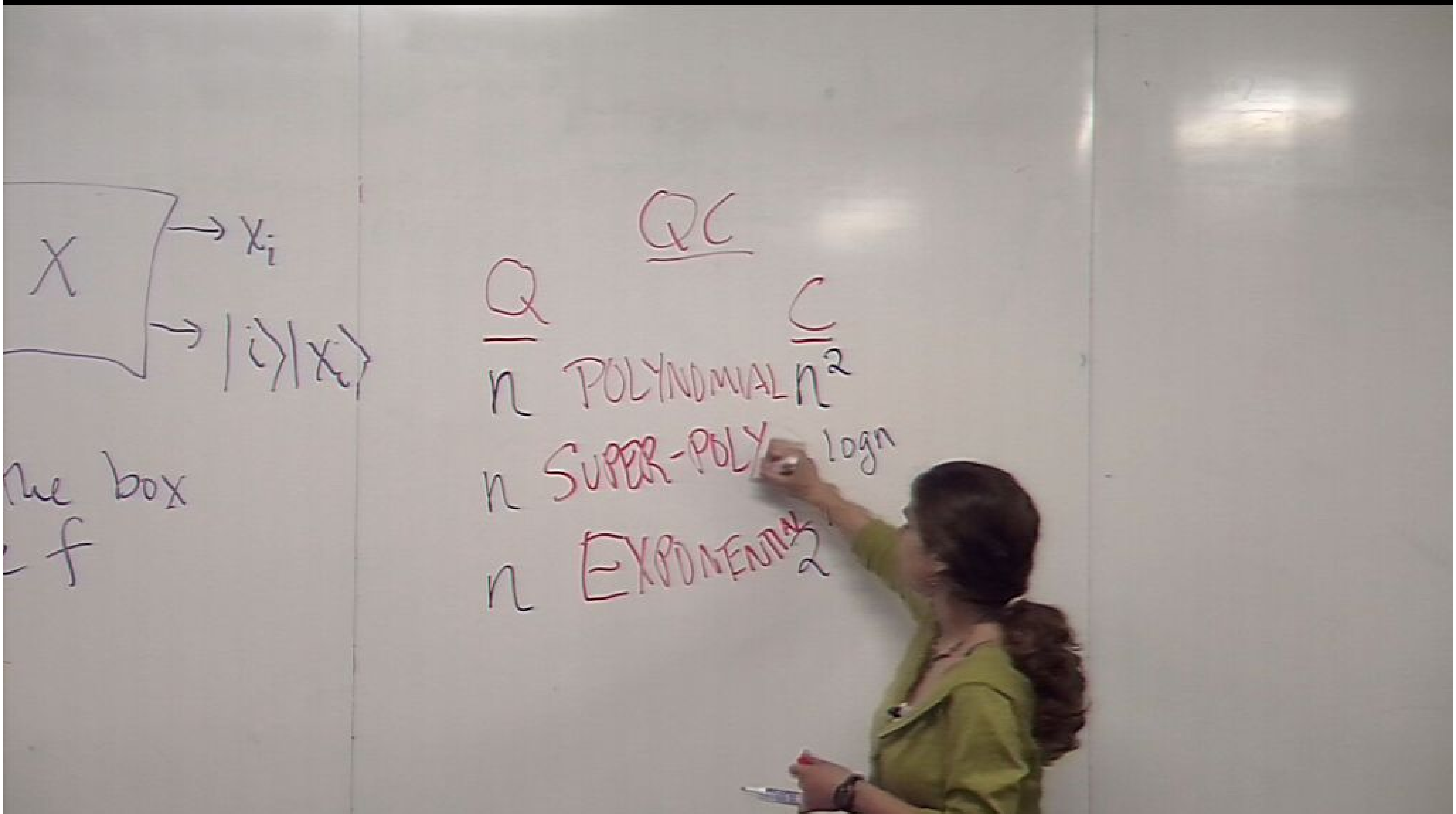
QC

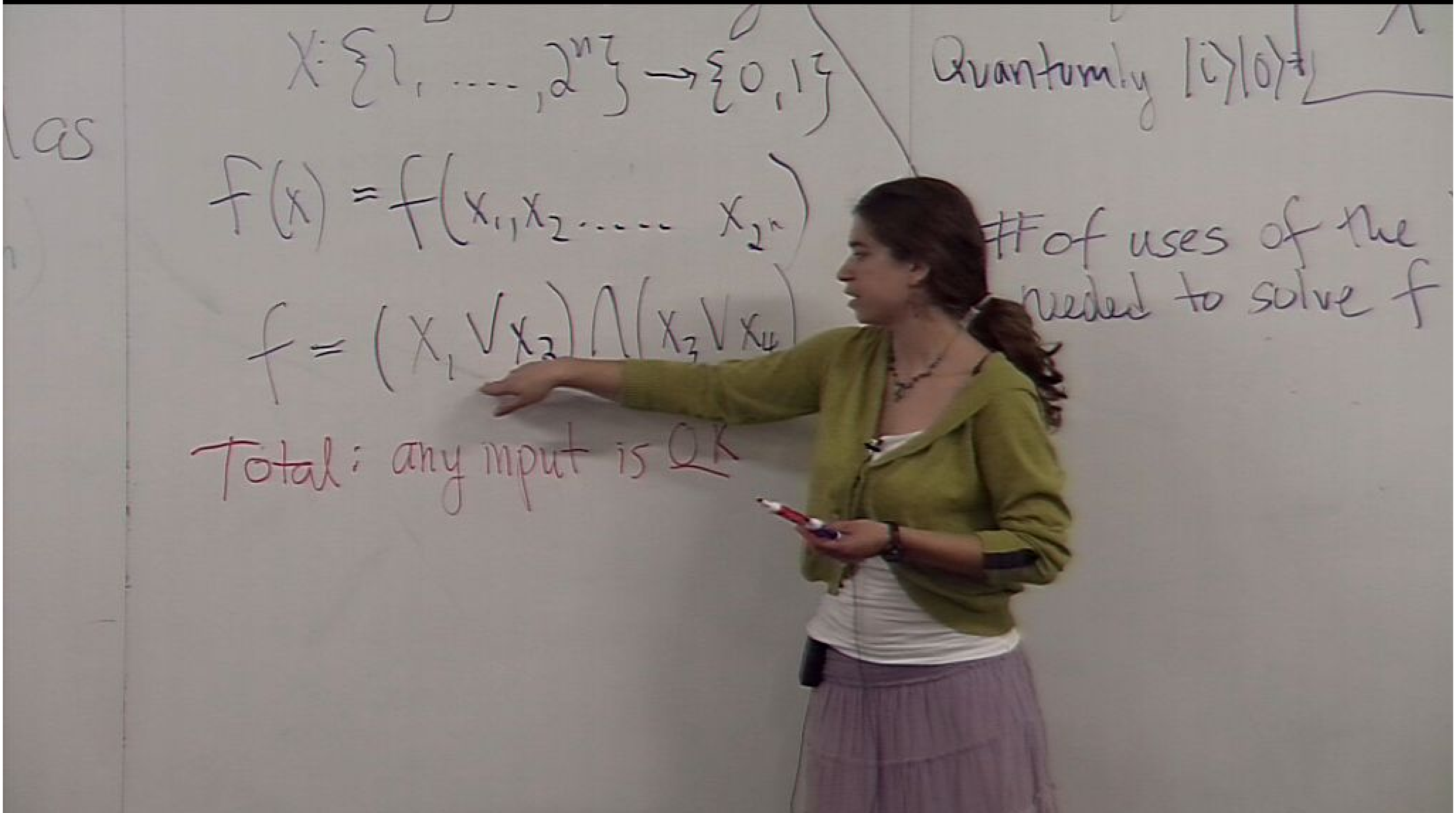
Q		C
n	POLYNOMIAL	n^2

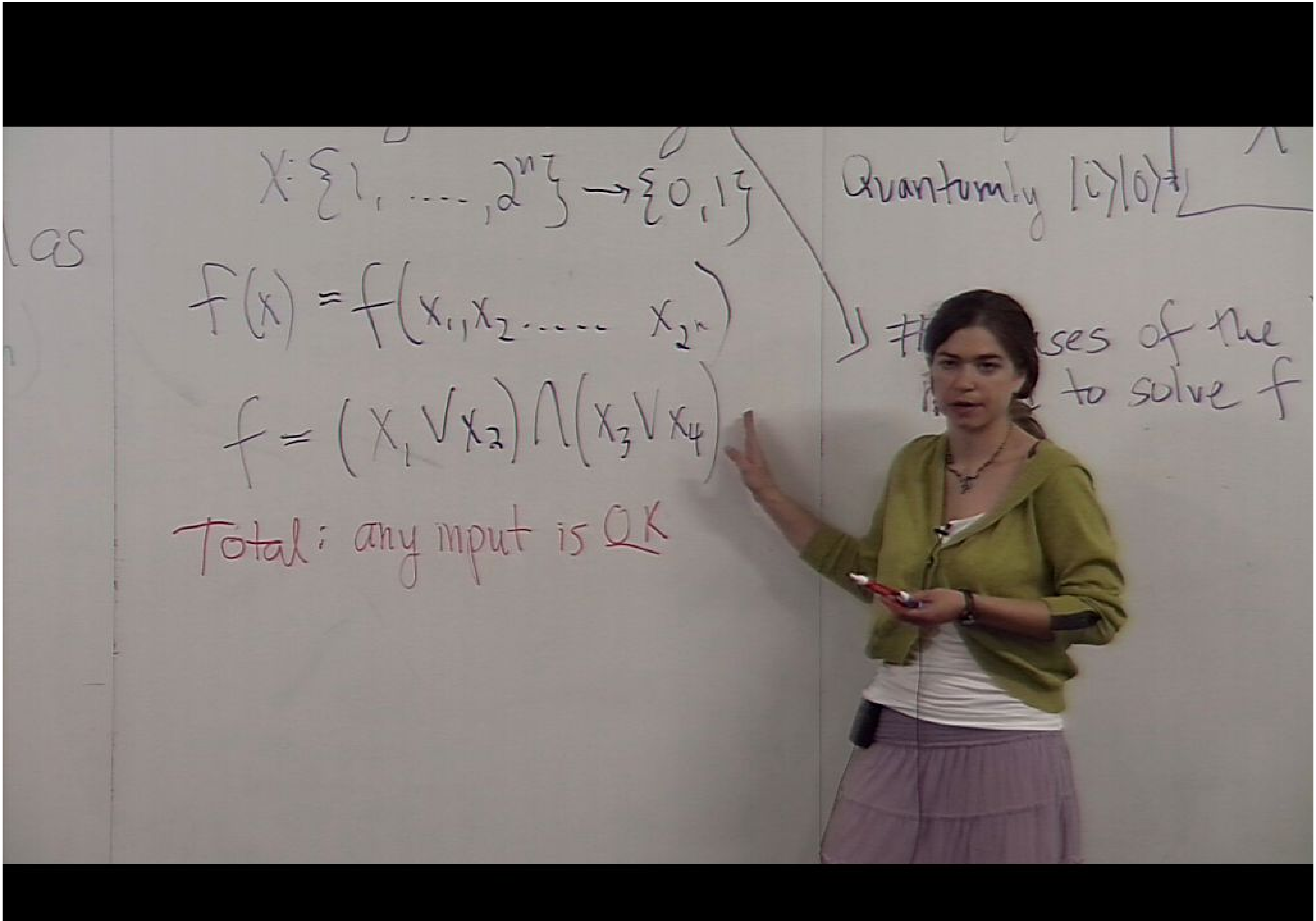
n


X
$\to x_i$
$\to |i\rangle|x_i\rangle$
he box
f
QC
Q
C
n POLYNOMIAL n^2
n
2^n

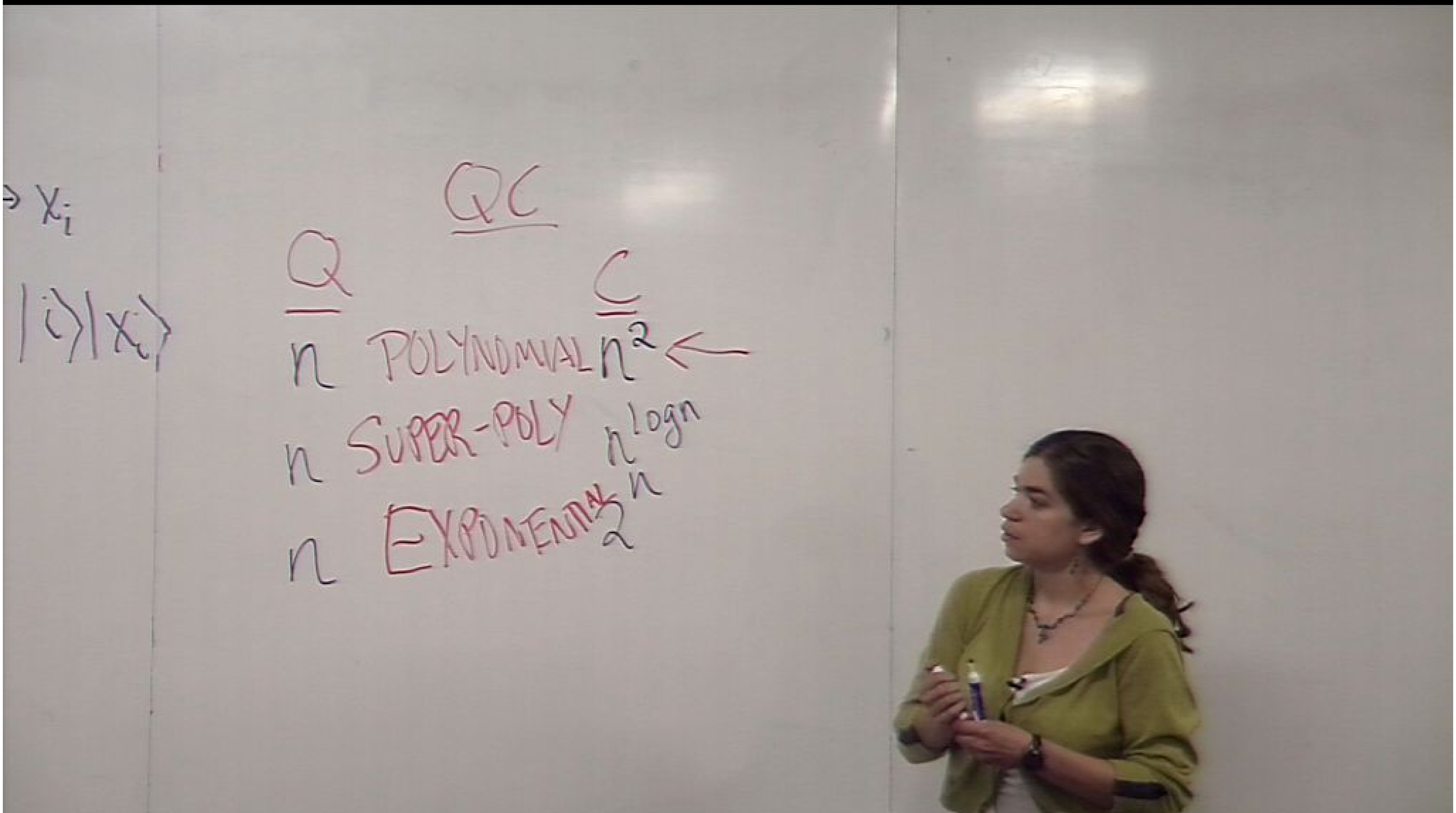

X
→ x_i
→ $|i\rangle|x_i\rangle$
he box
e f
QC
Q
C
n POLYNOMIAL n^2
n EXPONENTIAL 2^n


X
→ x_i
→ $|i\rangle|x_i\rangle$
the box
f
QC
Q
C
n POLYNOMIAL n^2
n SUPER-POLY
log n
n EXPONENTIAL
2

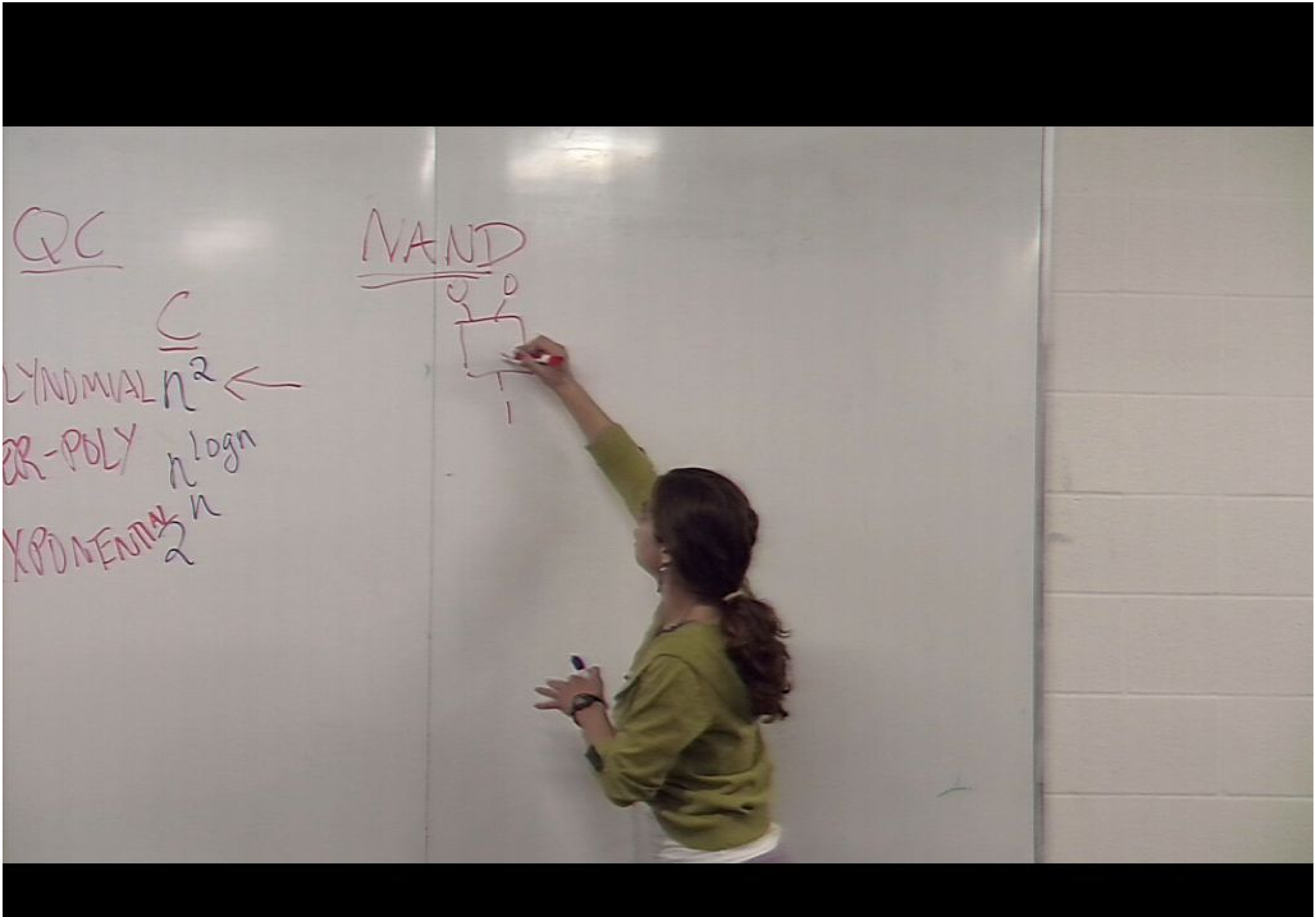

las
X: {1, ..., 2^n} → {0,1}
Quantumly |i⟩|0⟩ =
f(x) = f(x_1, x_2 ..... x_{2^n})
#of uses of the
needed to solve f
f = (x_1 ∨ x_2) ∧ (x_3 ∨ x_4)
Total: any input is OK

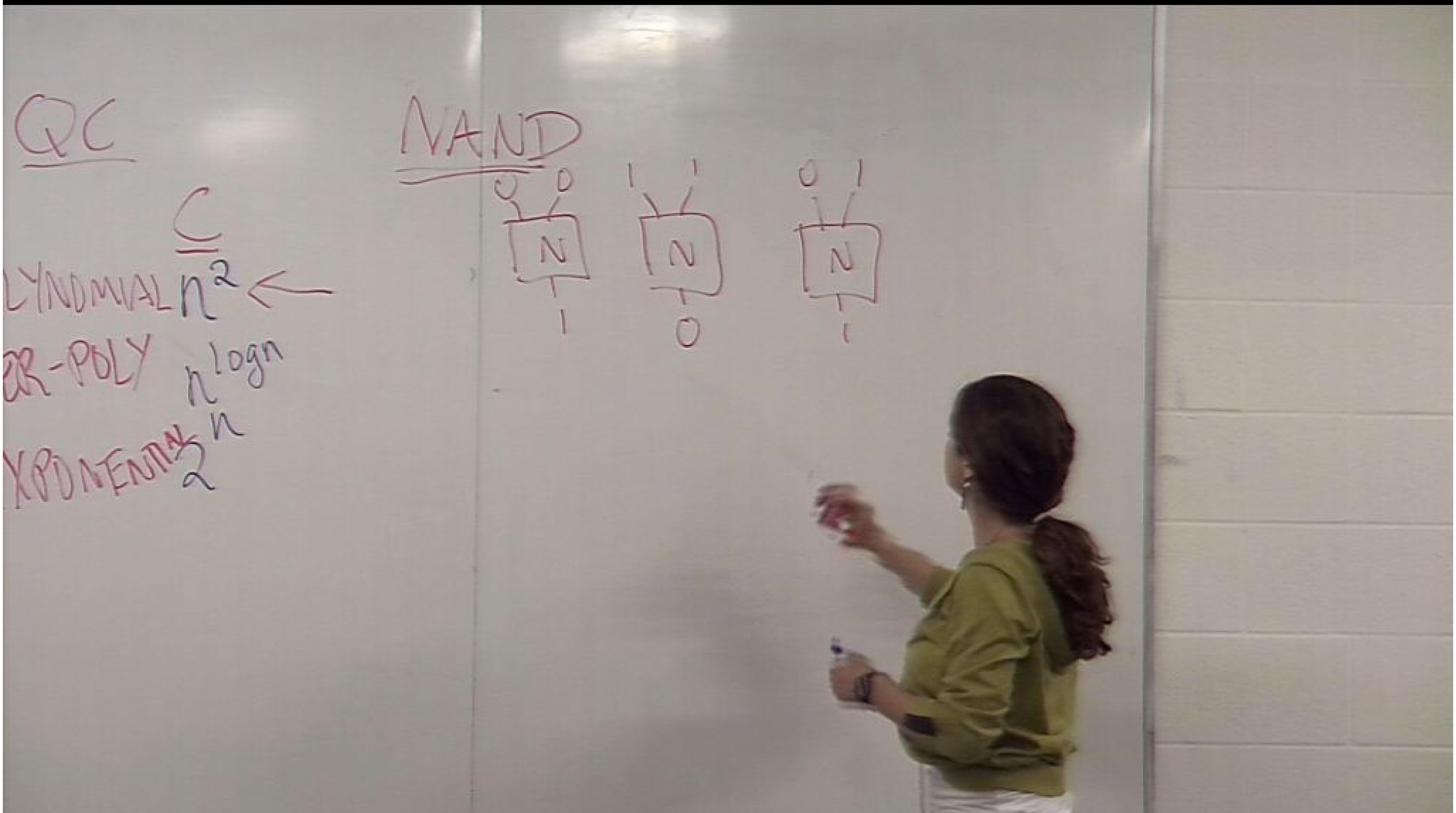


X: {1, ..., 2^n} → {0, 1}
Quantumly |i⟩|0⟩
f(x) = f(x1, x2 ..... x2^n)
f = (x1 ∨ x2) ∧ (x3 ∨ x4)
Total: any input is OK
to solve f

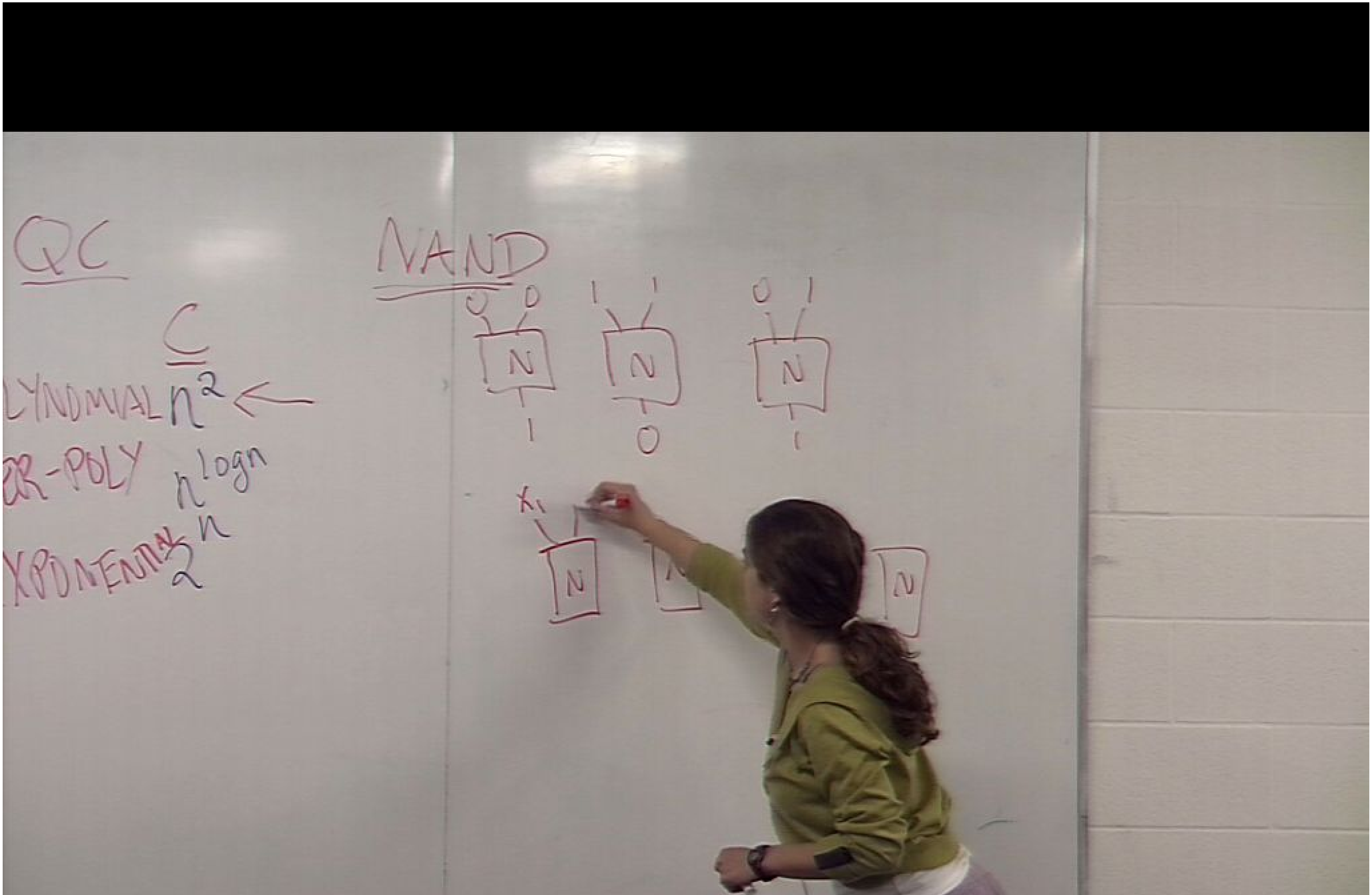


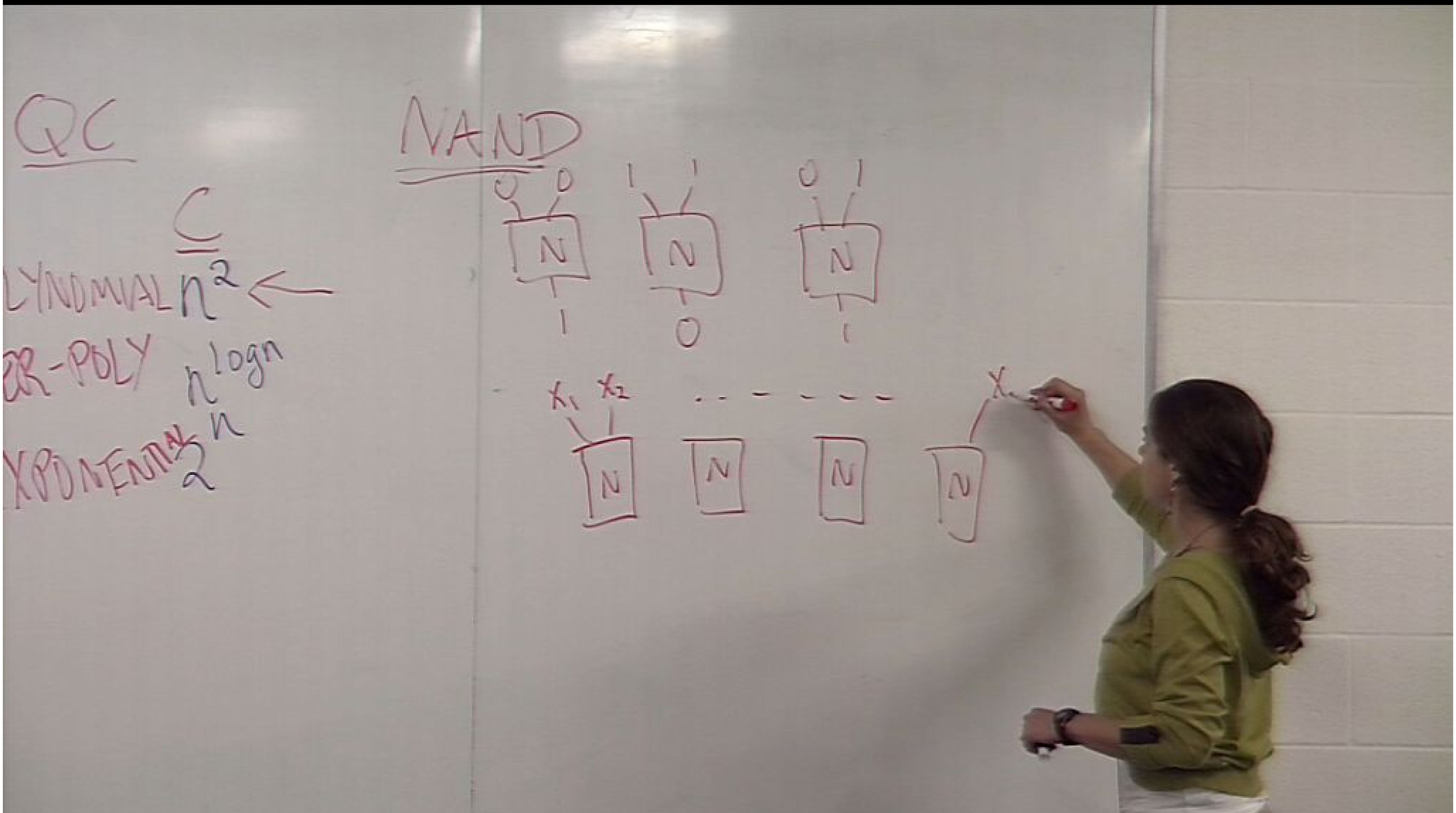
⇒ X_i
$|i\rangle|x_i\rangle$
QC
Q
C
n POLYNOMIAL n^2
n SUPER-POLY $n^{\log n}$
n EXPONENTIAL 2^n


QC
NAND
0 0
1
C
LYNOMIAL n²
ER-POLY n^logn
XPONENTIAL 2^n

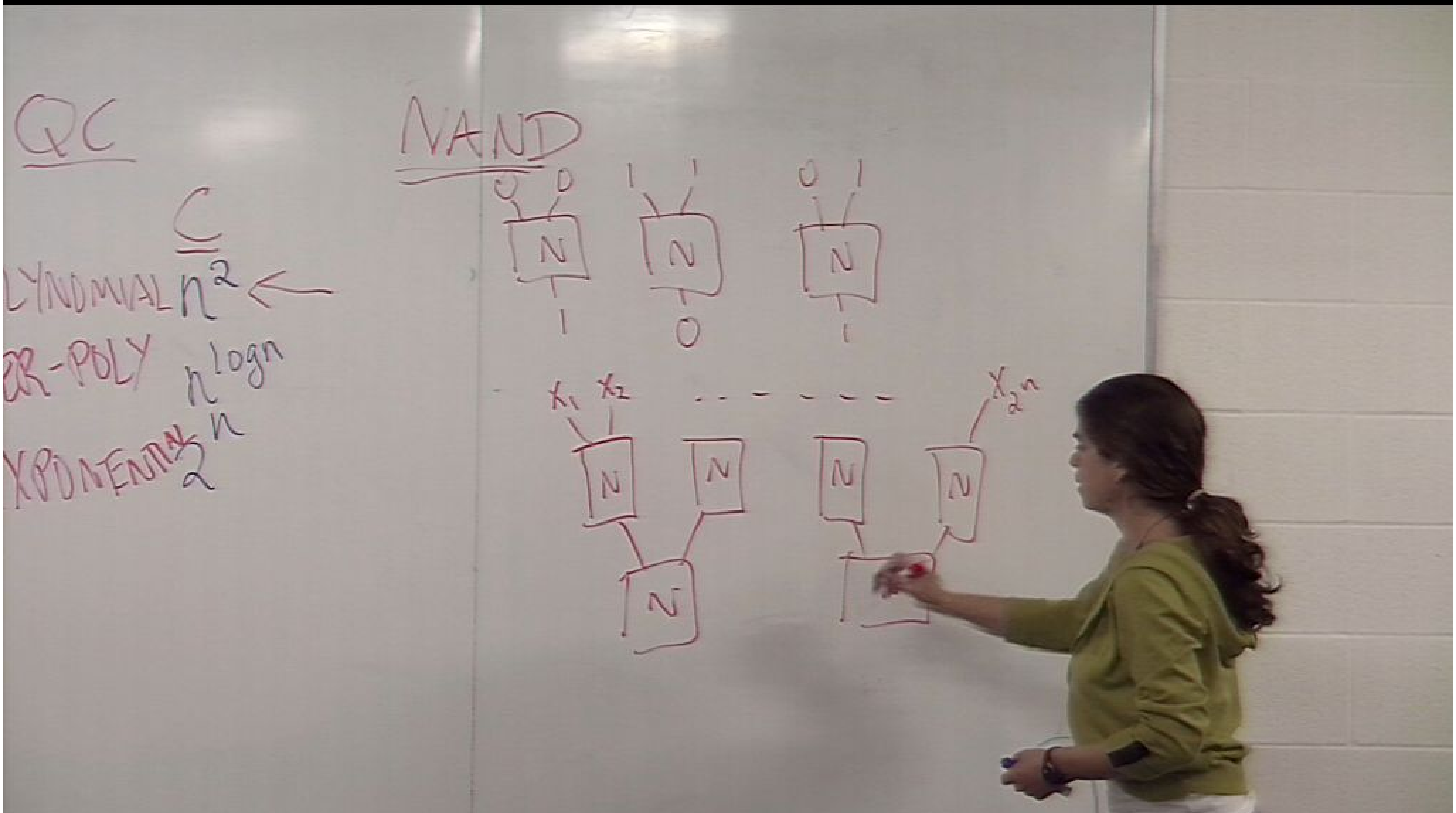


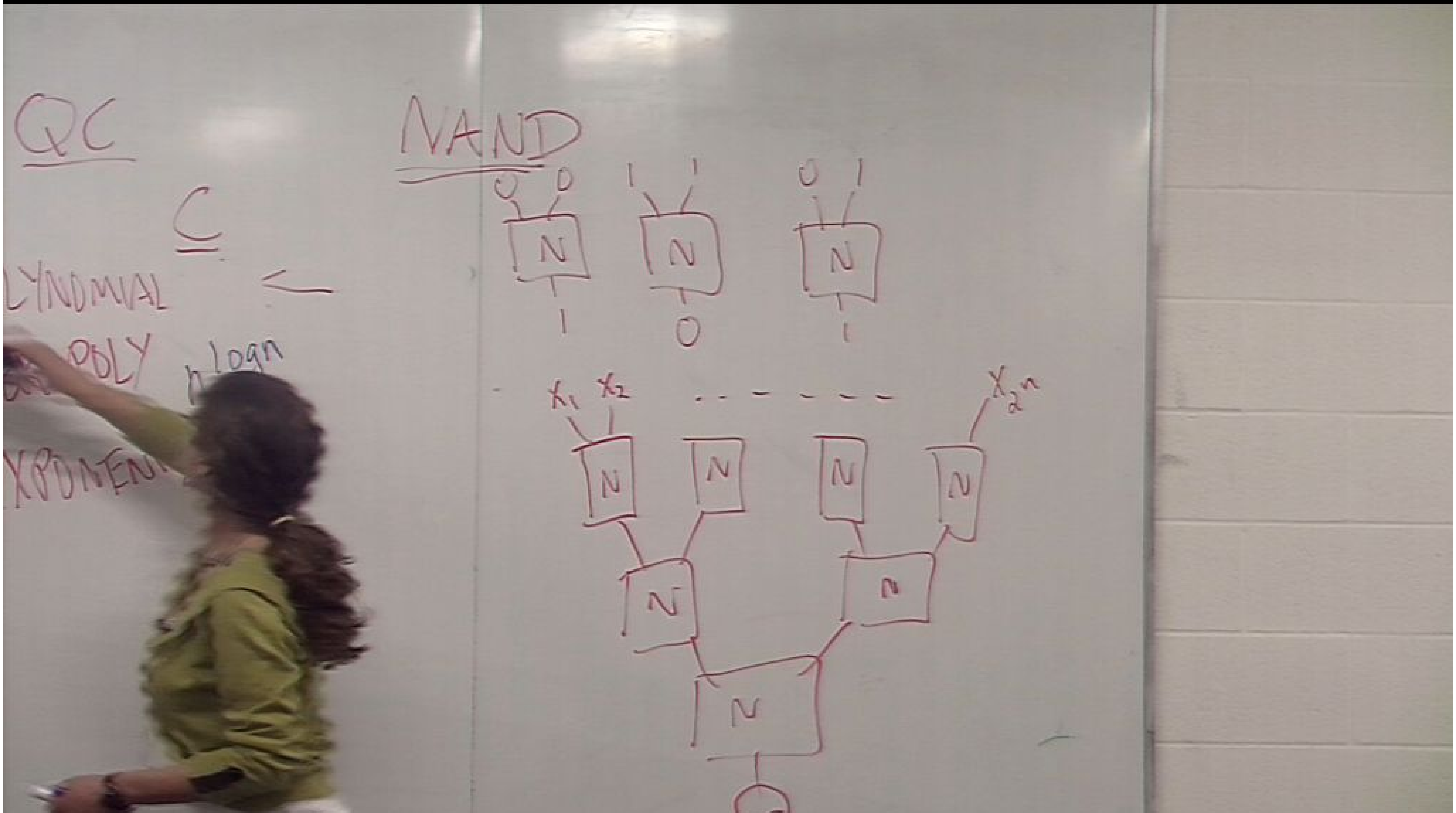
QC
C
LYNOMIAL n2
ER-POLY n logn
XPONENTIAL 2n
NAND
0 0
1 1
0 1
N
N
N
1
0
1


QC
C
LYNOMIAL n2
ER-POLY n logn
XPONENTIAL 2n
NAND
0 0
1 1
0 1
N
N
N
1
0
1
x1
N
N

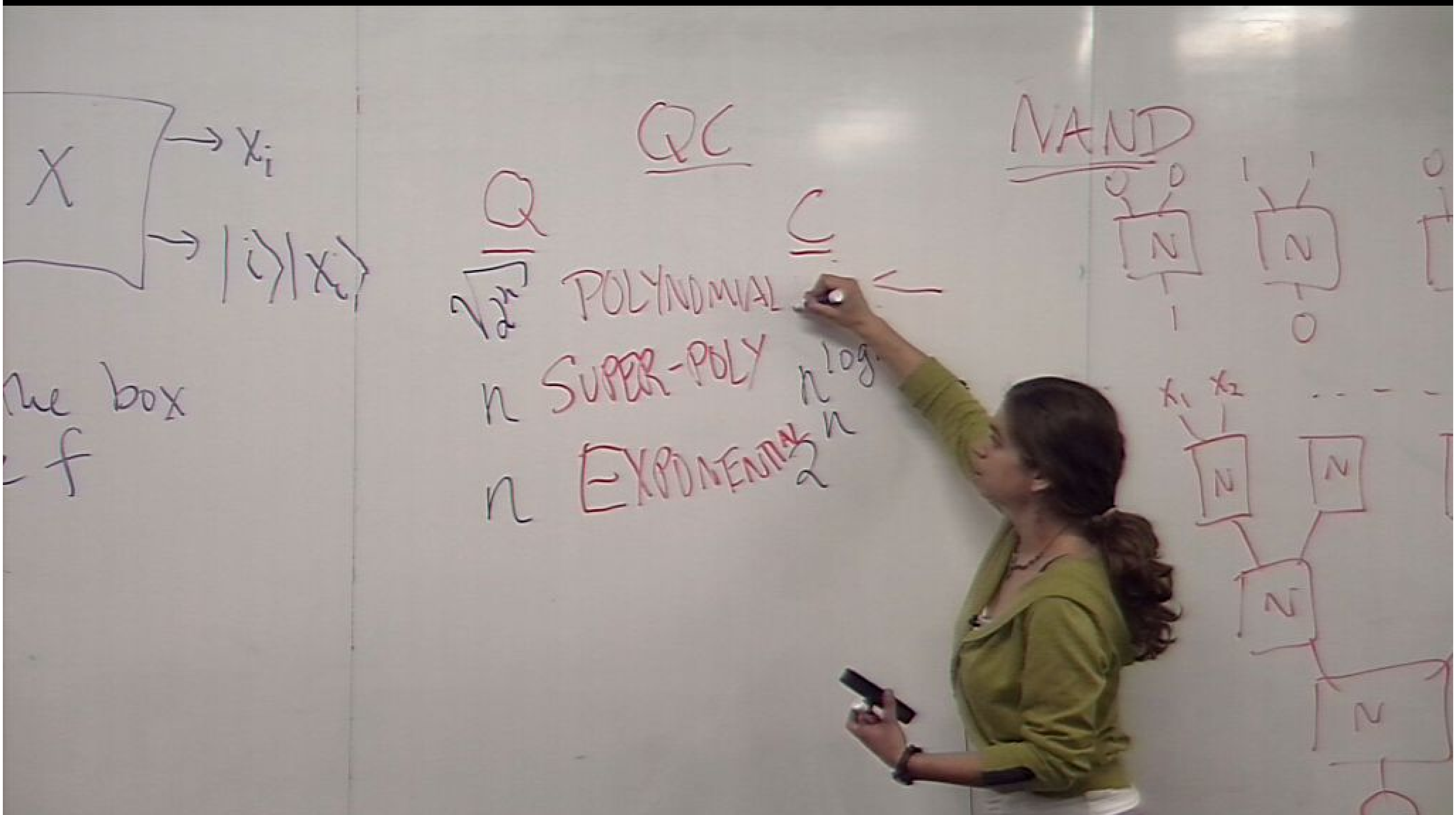


QC
NAND
C
LYNOMIAL n^2
ER-POLY n^log n
XPONENTIAL 2^n
0 0
1 1
0 1
N
N
N
1
0
1
x1 x2
N
N
N
N

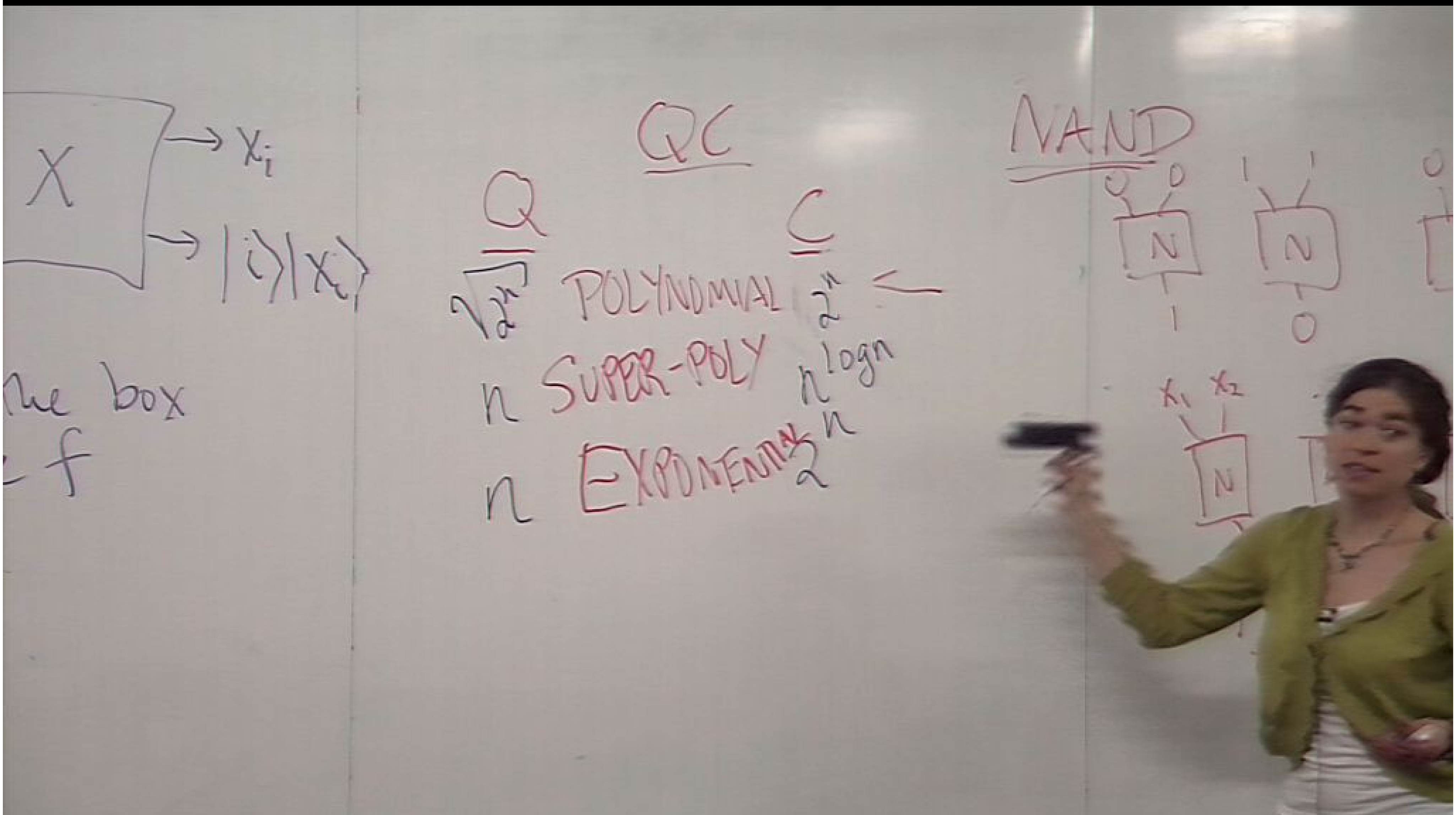

QC
C
LYNOMIAL n^2
ER-POLY $n^{\log n}$
XPONENTIAL 2^n
NAND
0 0
1 1
0 1
N
1
0
1
x_1 x_2
x_{2^n}

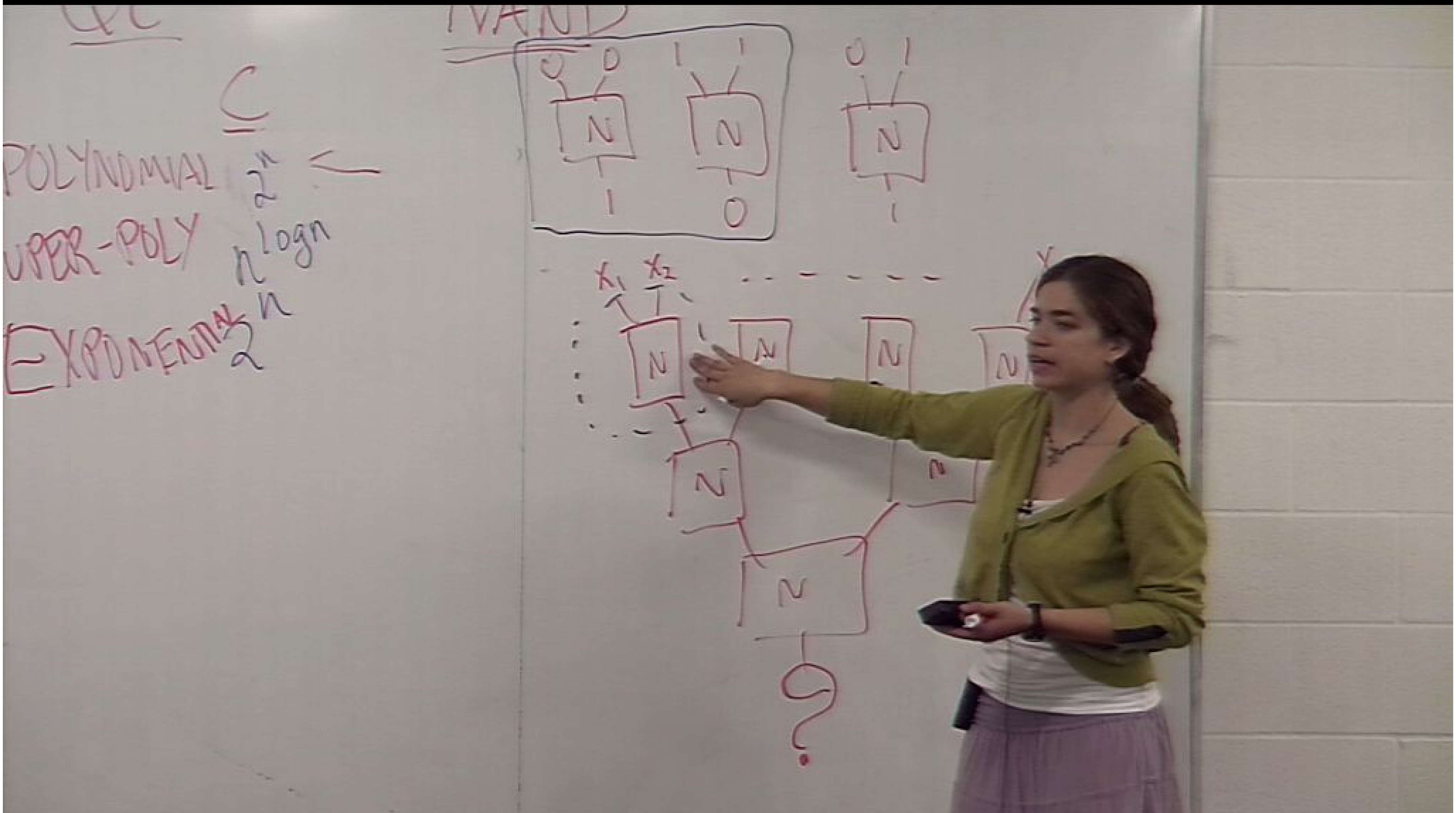


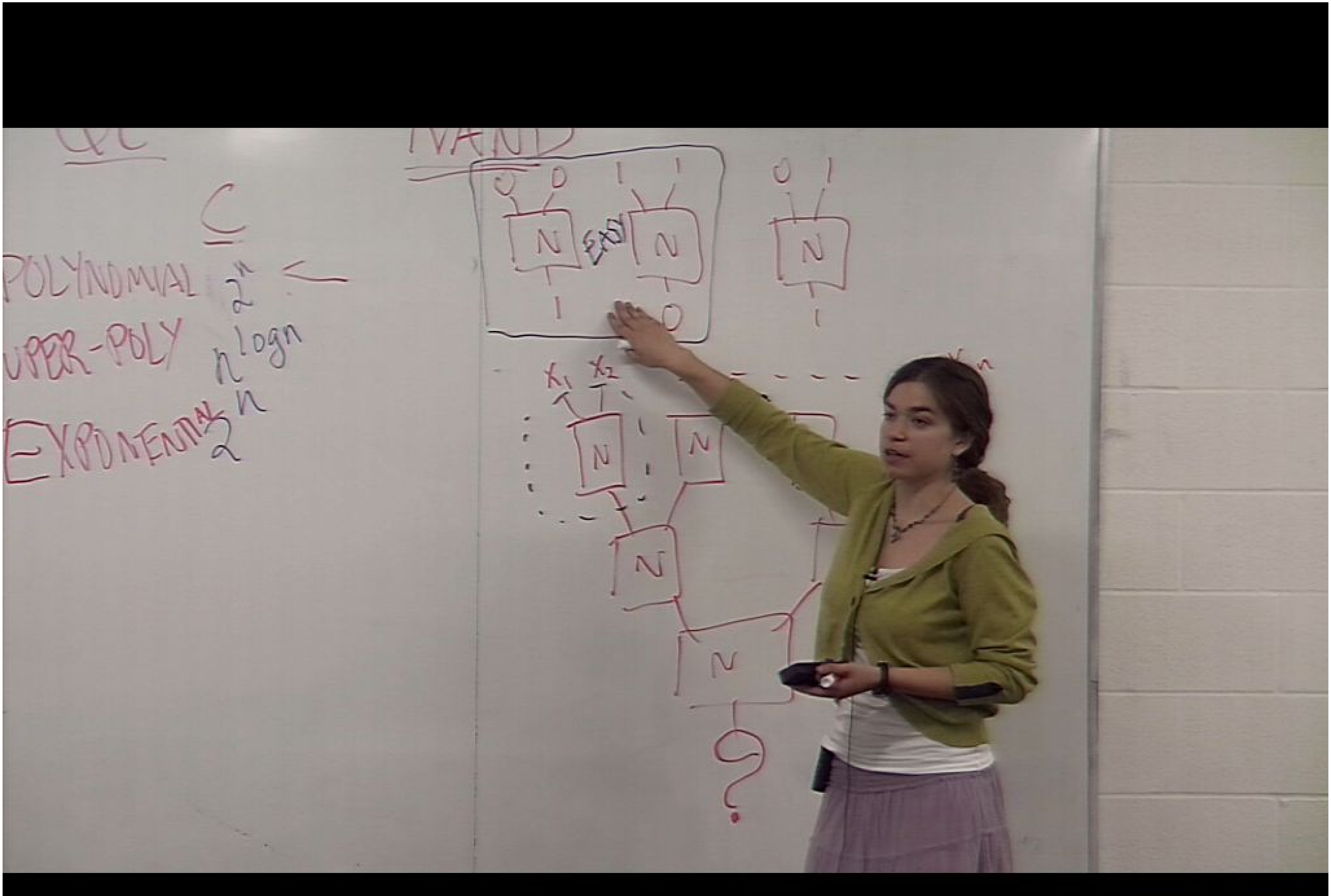
QC
NAND
LYNOMIAL
POLY
XPONEN
0 0
1 1
0 1
N
1
0
1
X1 X2
X2n
N

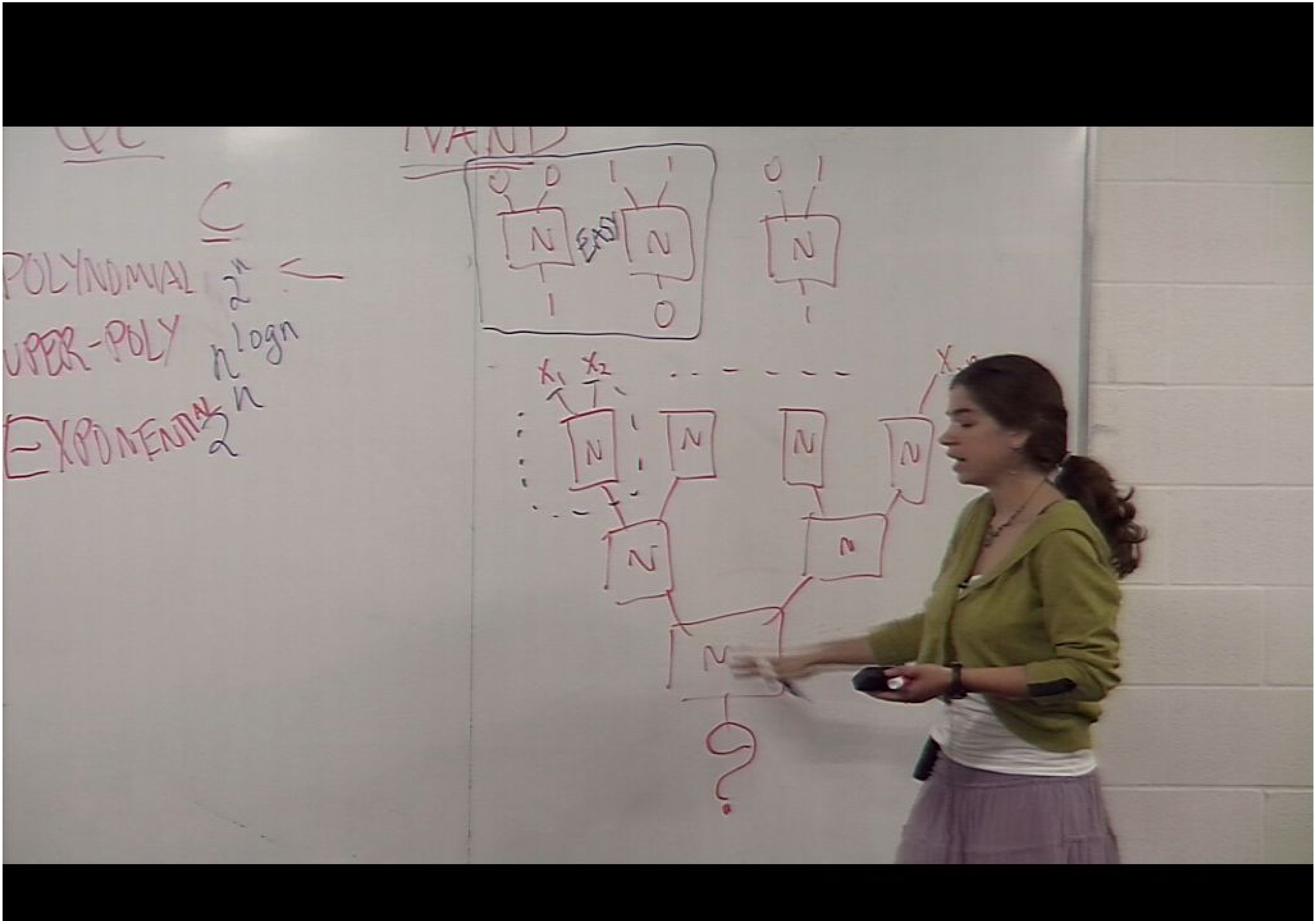


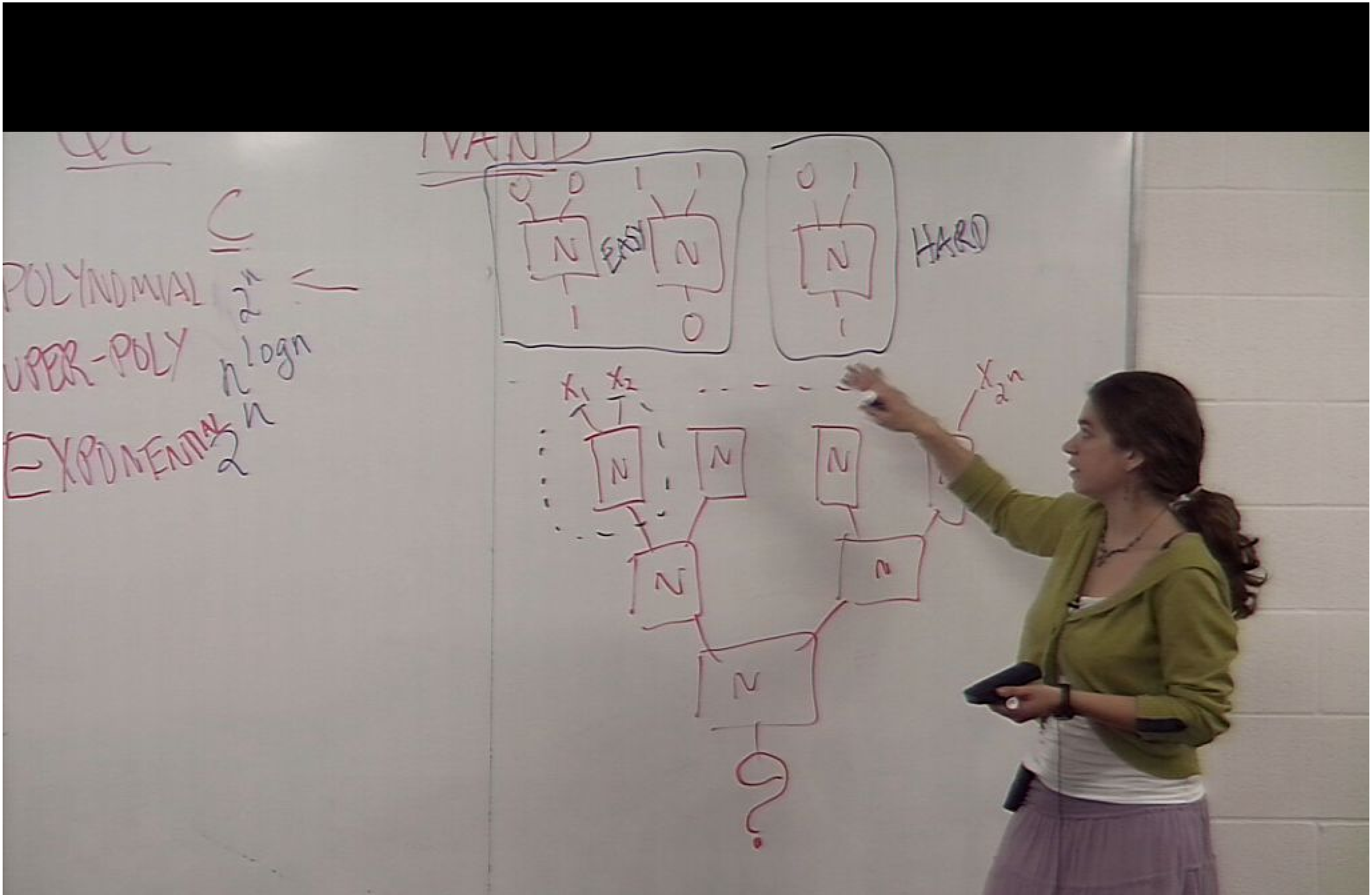
X
→ x_i
→ $|i\rangle|x_i\rangle$
he box
f
QC
Q
C
$\sqrt{2^n}$ POLYNOMIAL
n SUPER-POLY $n^{\log}$
n EXPONENTIAL 2^n
NAND
N
N
x_1 x_2
N
N
N
N


X
$\to x_i$
$\to |i\rangle|x_i\rangle$
the box
f
QC
Q
C
$\sqrt{2^n}$ POLYNOMIAL 2^n
n SUPER-POLY $n^{\log n}$
n EXPONENTIAL 2^n
NAND
N
N
N
x_1 x_2

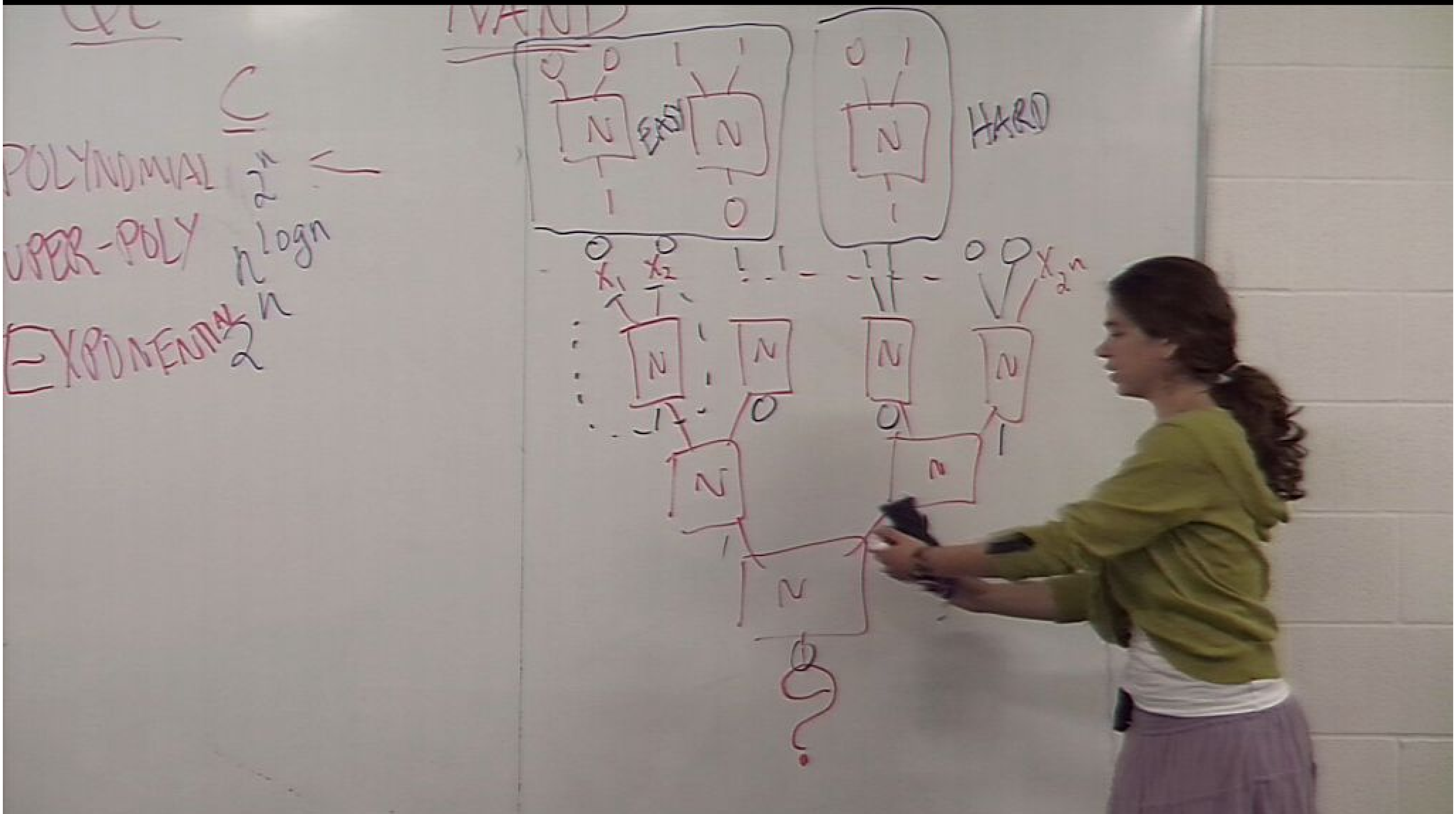

NAND
POLYNOMIAL
UPER-POLY
EXPONENTIAL
2^n
n^log n
2^n
N
x1
x2
?


NAND
POLYNOMIAL
UPER-POLY
EXPONENTIAL
EASY
N
x_1 x_2
2^n
$n^{\log n}$
?


NAND
POLYNOMIAL
2^n
UPER-POLY
n^log n
EXPONENTIAL 2^n
0 0
1 1
0 1
N
EASY
1
0
X1 X2
?



NAND
EASY
HARD
POLYNOMIAL
UPER-POLY
EXPONENTIAL
N
x1 x2
x2^n
?


NAND
POLYNOMIAL
UPER-POLY
EXPONENTIAL
EASY
HARD
N