LECTURE

4/11/23

→ RUDRATA CYCE is NP-Complete

→ Every problem in NP $\leq_p$ CIRCUITSAT

→ CIRCUITSAT $\leq_p$ 3SAT

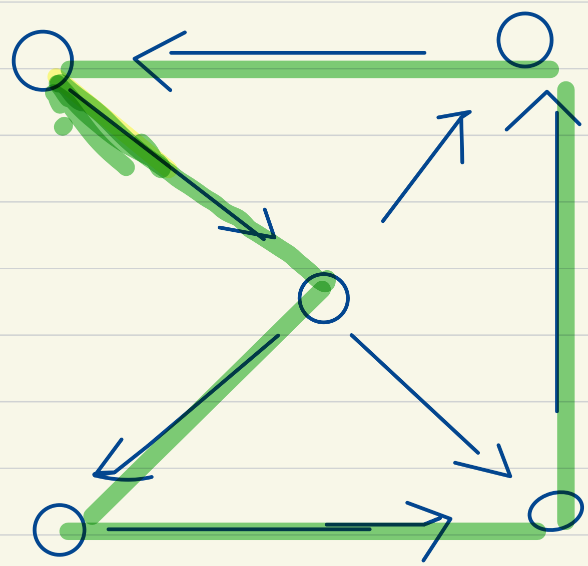
→ APPROXIMATION ALGOS.

(DIRECTED)

RUDRATA CYCLE

INPUT: A directed Graph G

SOL: A directed cycle passing through every vertex exactly once



THEOREM: RUDRATA CYCLE is NP-Complete

PROOF:

1) RUDRATA CYCLE $\in$ NP

(exercise)

(3SAT)

2) SOME NP-Complete Problem $\leq_p$ RUDRATA CYCLE

3SAT Problem

INPUT: A 3-SAT Formula

$(x \vee y \vee \bar{z}) \wedge (z \vee w \vee p) \wedge \ldots$

SOL: An assignment to variables satisfying

3SAT $\leq_P$ RUDRATA CYCLE

3SAT instance: Φ

$(x \vee y \vee z) \wedge$

$(\bar{y} \vee z \vee w) \wedge$

$(\bar{x} \vee y \vee \bar{w})$ ...

" Reduction

Algorithm "

RUDRATA CYCLE G

Input: Directed Graph $G=(V,E)$

Sol:

SUCH THAT:

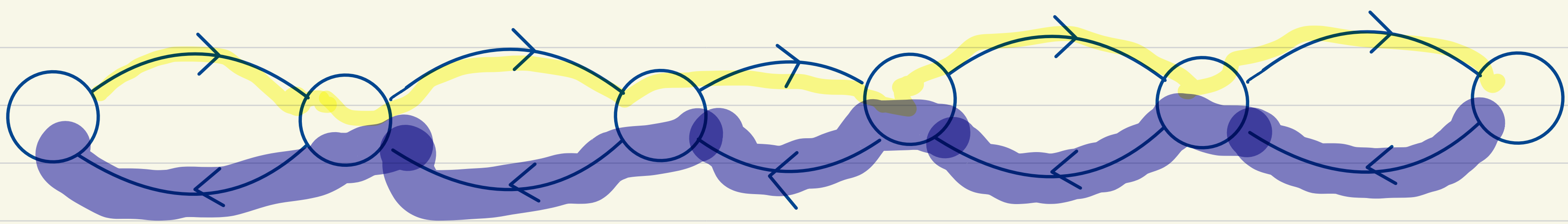
$\exists$ a satisfying assignment to formula Φ

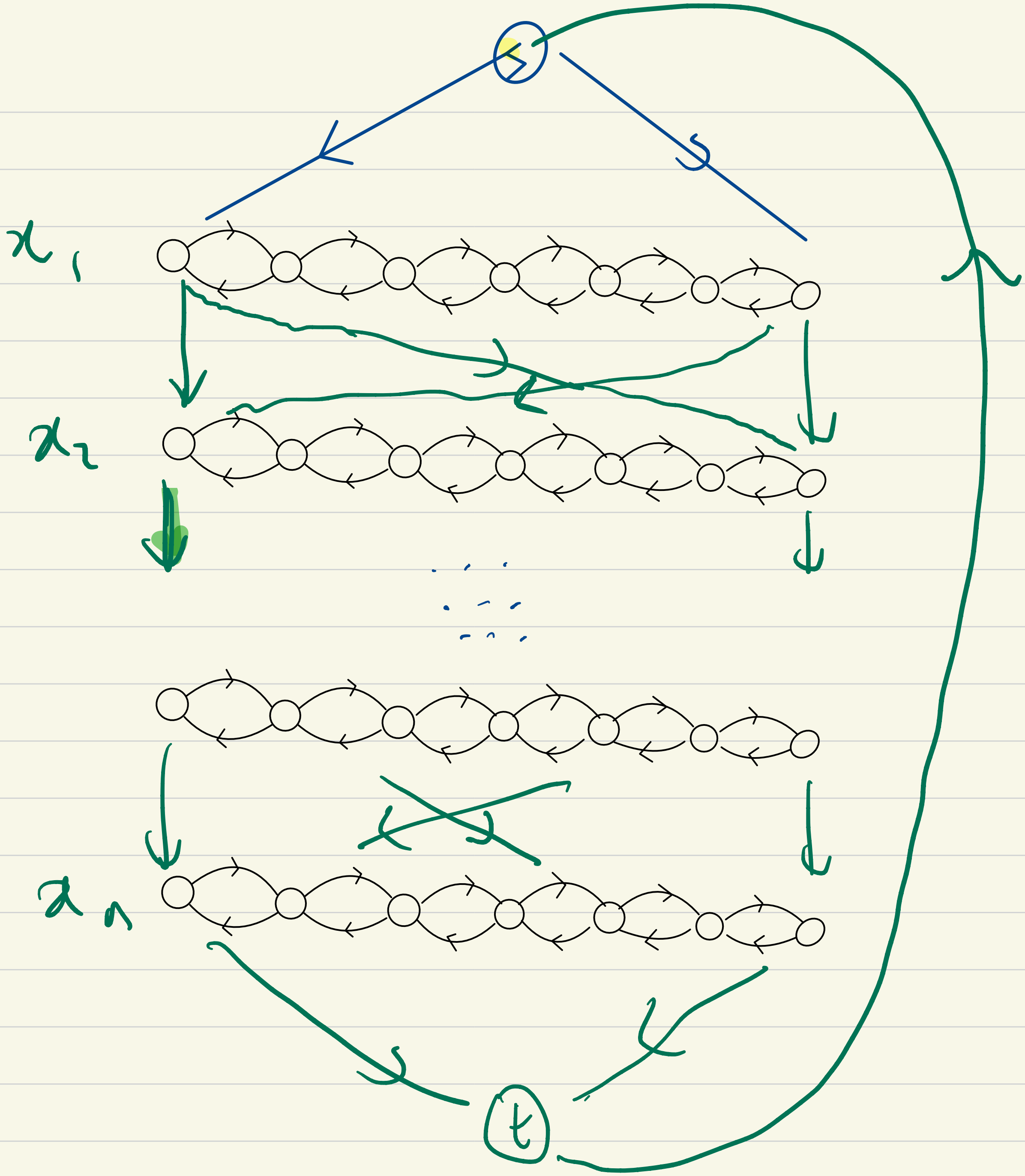
$\iff$ $\exists$ a Rudrata cycle in graph G.

INTUITION :

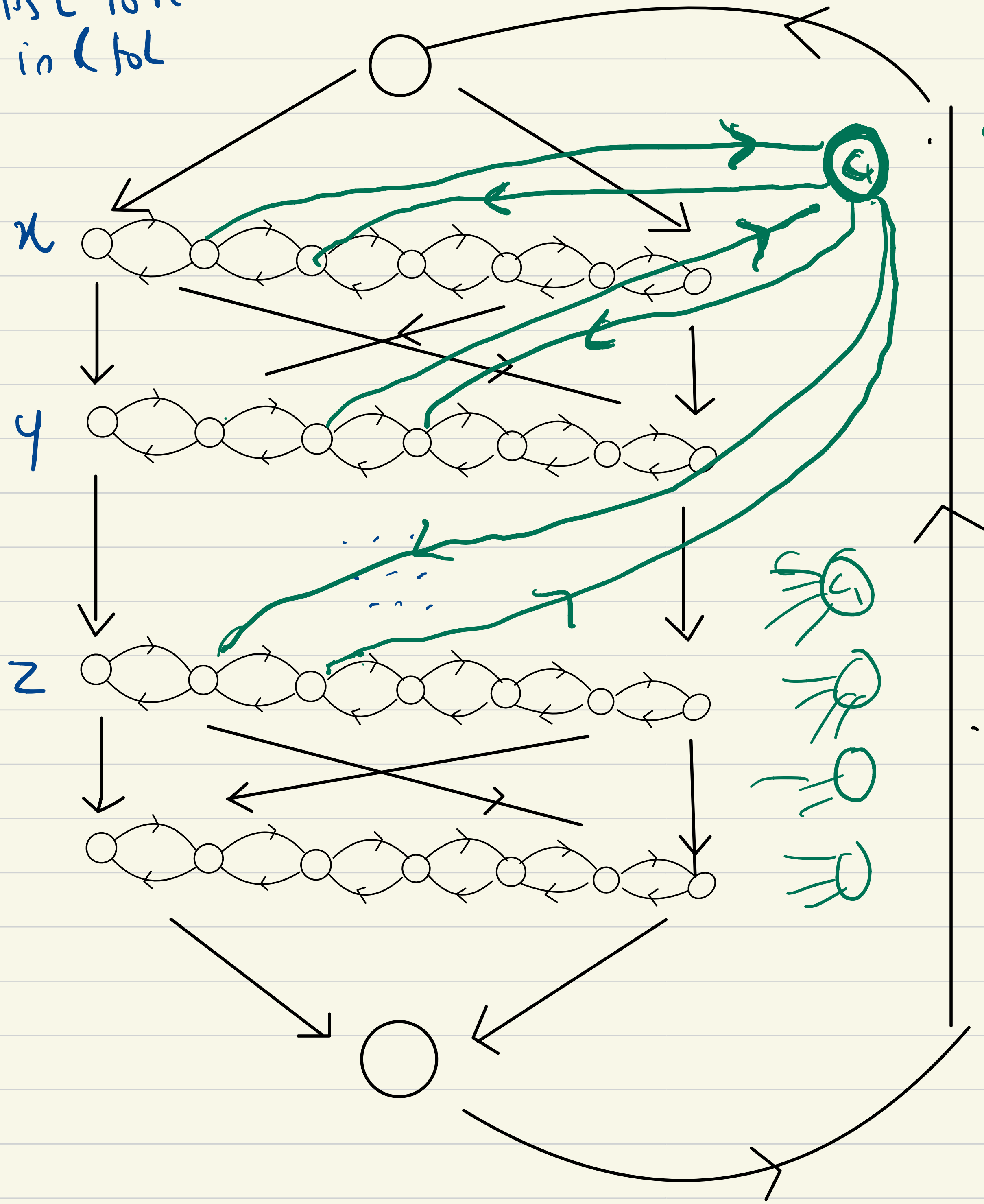
$x \in \{0, 1\}$

1 → Left to Right
OR
0 → Right to Left




s
x_1
x_2
x_n
t

1 in L to R
0 in R to L



$C_1 = (x \vee y \vee \bar{z})$

C_2

$\Updownarrow$

x in L to R
OR
y in L to R
OR
z is R to L

Clause C can be visited if ONLY IF tour is going L to R on x

OR is going L to R on y

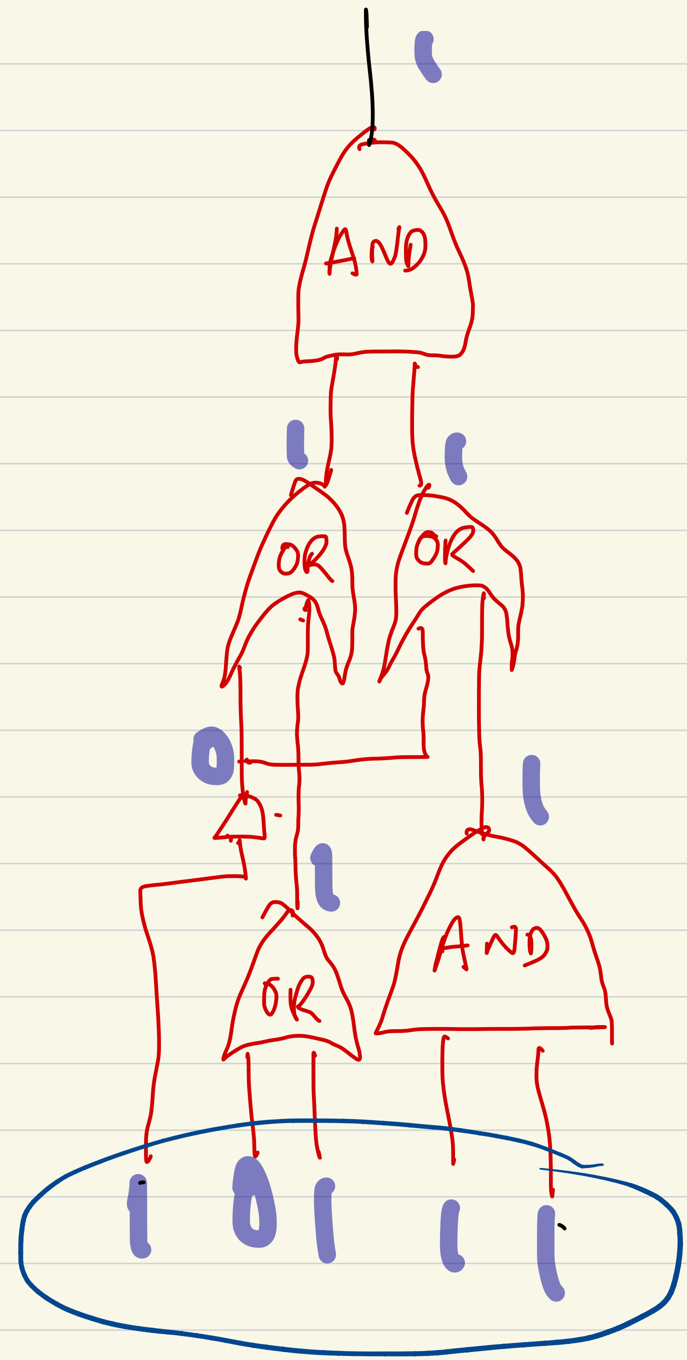
OR is going R to L on z

CIRCUIT SAT

INPUT: 1) Circuit with AND/OR/NOT gates

2) n inputs

SOLUTION: An boolean assignment so that output = 1.



THEOREM:

EVERY PROBLEM IN NP $\leq_p$ CIRCUIT SAT.

PROOF: FOR CONCRETENESS

Consider FACTORIZATION

FACTORIZATION

INPUT: An $\underline{n}$ bit number N

SOL: p, q $\quad p, q > 1$

and $p \cdot q = N$

$\le_P$

CIRCUIT SAT

INPUT: Circuit C

SOL: x s.t $C(x) = 1$

BASIC IDEA:

NP problems have verification algos

All computation can be done on circuits

Factorisation $\leq_p$ Circuit SAT:

Proof. Factorization has a verification algo

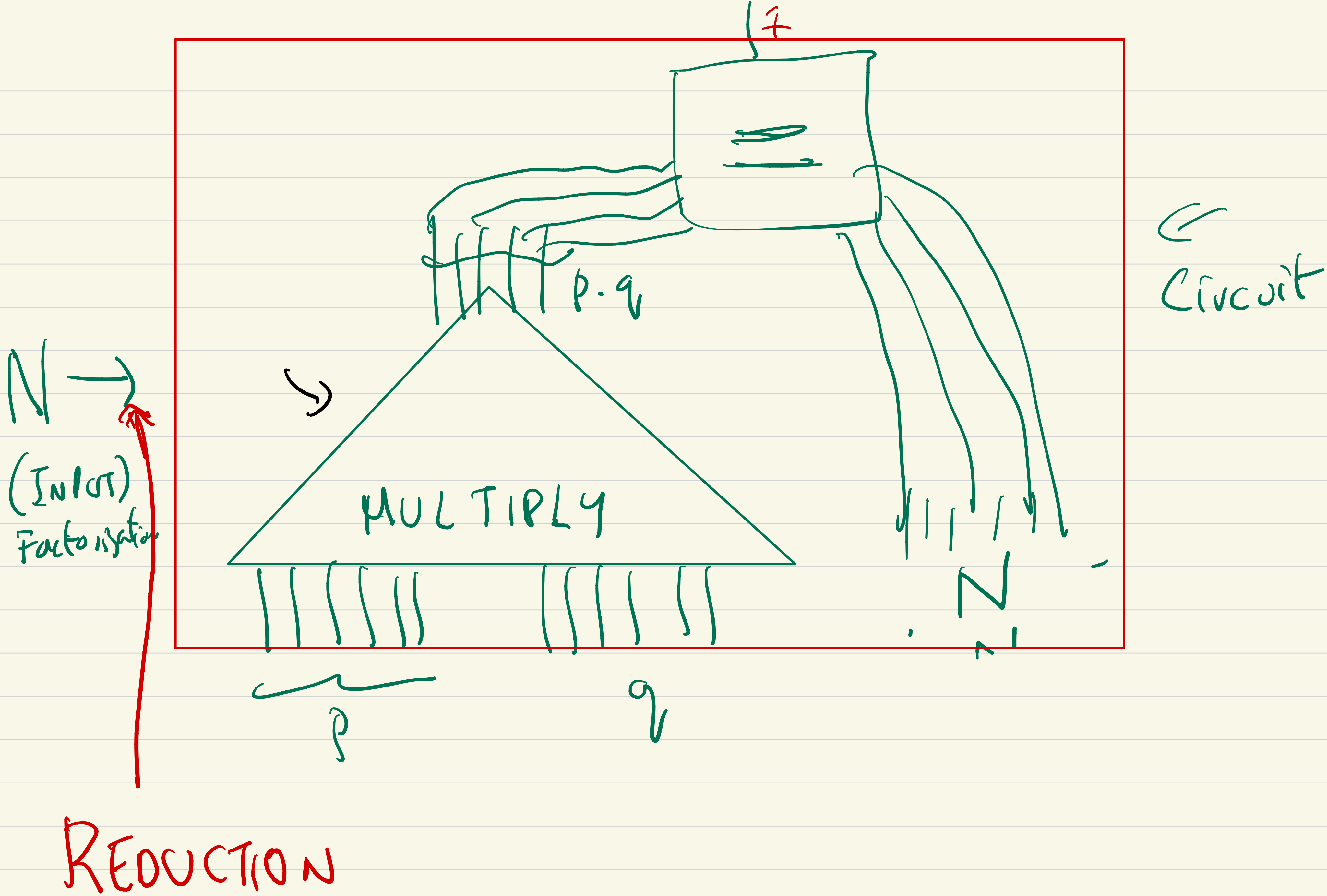
VERIFY (INPUT Number N , SOLUTION Numbers p, q)

{

Check if $p \cdot q = N$

AND $p > 1$

}


=
p·q
MULTIPLY
p
q
N
Circuit
N →
(INPUT)
Factorisation
REDUCTION

CIRCUIT SAT

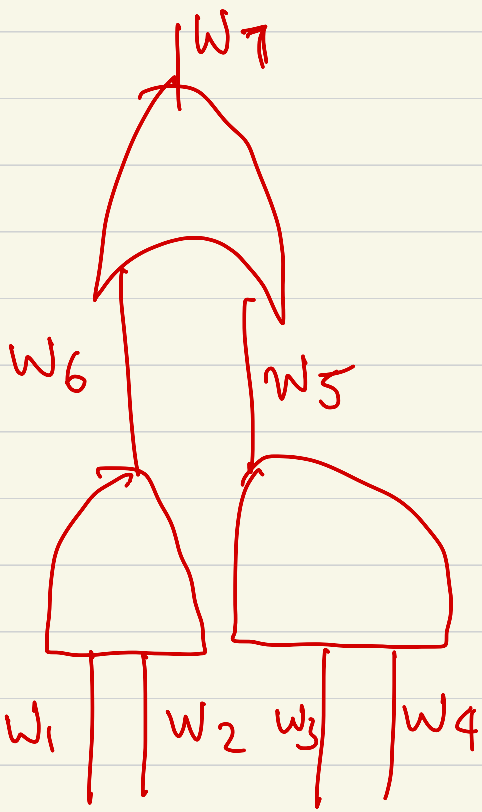
INPUT: A circuit C

SOL: An assignment x, s.t $C(x) = 1$

3SAT

INPUT: A 3SAT FORMULA

SOL: A satisfying assignment.



(Assigned Readings)

INDSET:

INPUT: Graph $G=(V,E)$
integer K

SOL: An ind. set of size K

CLIQUE

INPUT: Graph $G'=(V',E')$
integer K

SOL: A clique of size K

INDSET:

INPUT: Graph $G=(V,E)$
integer K

SOL: An ind. set of size K

VERTEX COVER

INPUT: Graph $G'=(V',E')$
integer ℓ

SOL: A vertex cover of size ℓ

APPROXIMATION ALGORITHM

Def: For A minimisation problem P, $(\alpha > 1)$

an algorithm is an α-approximation algorithm

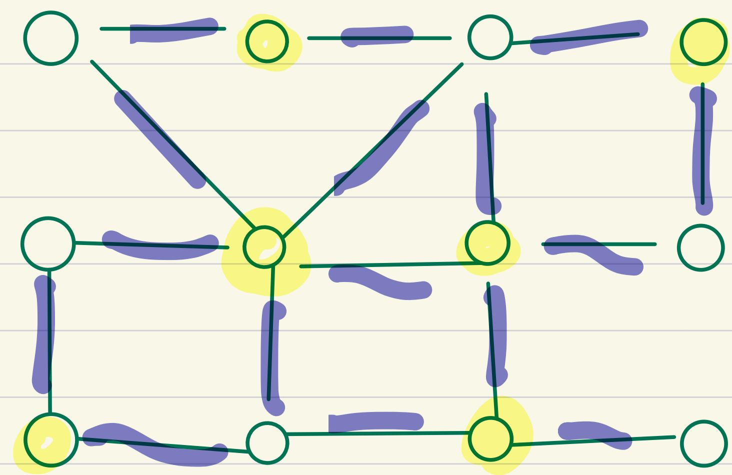
if $\forall$ input I from P

$$\text{ALG-OUTPUT}(I) \leq \alpha \cdot \text{OPT}(I)$$

[For maximization problem

$$\text{ALG OUTPUT}(I) \geq \text{OPT}(I)/\alpha$$]

MINIMUM VERTEX COVER $\in$ NP-hard.



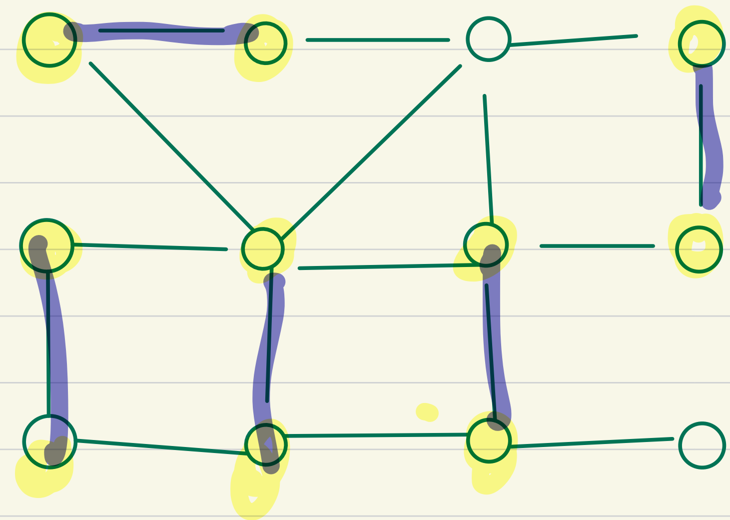
INPUT: Graph $G=(V,E)$

SOL: A vertex cover $S \subseteq V$ of smallest size

Definition: S is a vertex cover if every edge is covered by S

i.e $(u,v) \in E \Rightarrow u \in S$ or $v \in S$ or both

A 2-APPROXIMATION ALGO



→ Pick a maximal matching M

(maximal matching: keep adding edges until one can't)

Purple edges: Maximal Matching

→ Output $S = \{$ Both endpoints of edges in M $\}$ (YELLOW)

FACT I: S is a vertex cover.

PROOF: M is a maximal matching

Every edge overlaps some edge in M.

$\Rightarrow$ Every edge overlaps some $\underbrace{\text{endpoint of } M}_{\text{a vertex in } S}$.

$\rightarrow |S| = 2\,|\text{Maximal matching } M|$

$\boxed{\text{Fact: } |\text{OPTIMAL VERTEX COVER}| \geq |\text{Maximal Matching } M|}$

Proof: Every vertex cover has ATLEAST ONE VERTEX PER EDGE OF M.

$$|OPT| \geq |M|$$

$$|S| = 2 \cdot |\text{Maximal Matching}|$$

$$\leq 2 \cdot |\text{OPTIMAL VERTEX COVER}|$$