Lecture 15: [NP Completeness] and [Approximations]

- P vs NP
- blah "unless P = NP"

  unless something weird happens.

- NP-complete ("hardest problems in NP")

operational defn

Q is NP complete

if (1) Q ∈ NP

(3) 3SAT is NP-complete

(2) (∃ another NP-complete problem Q' st

$Q \in P \Rightarrow Q' \in P$

if Q ∈ P

then NP = P

(1) Independent Set / Stable Set (

INDSET

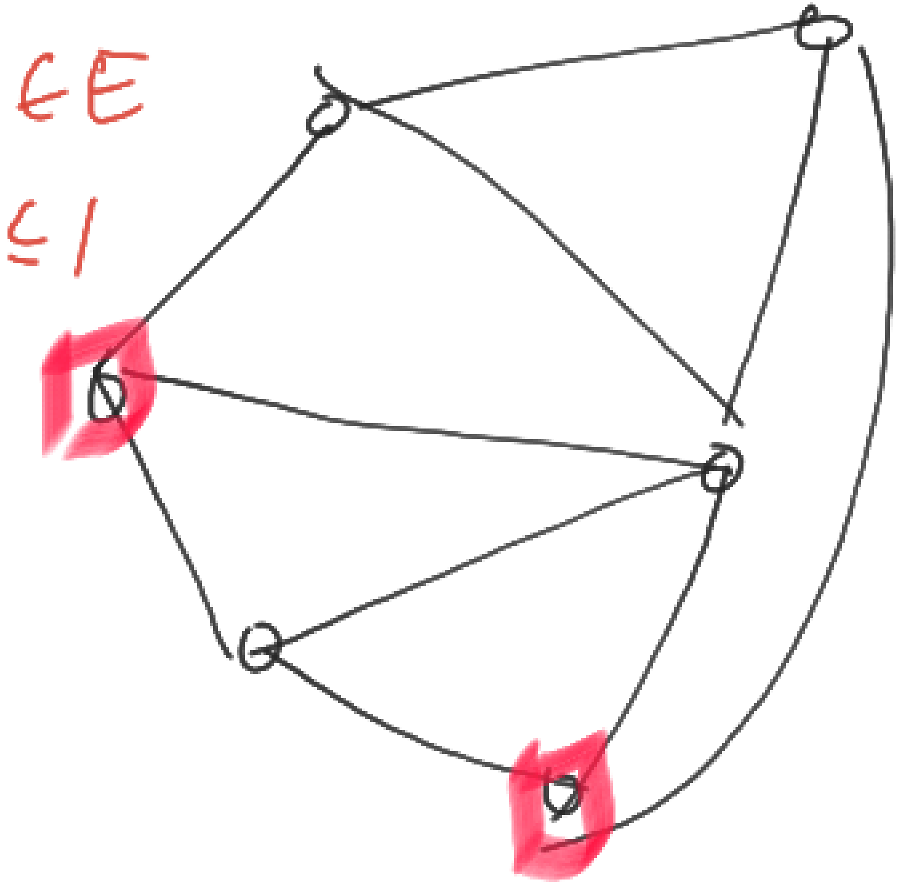
Input: graph G, number K

does $\exists$ indset in G of size $\geq K$

Ind set: $S \subseteq V$

st $\forall (u,v) \in E$

$|S \cap \{u,v\}| \leq 1$



Claim: INDSET is NP-complete

(1) Indset $\in$ NP ✓

(2) "reduce an NP hard problem to Indset"

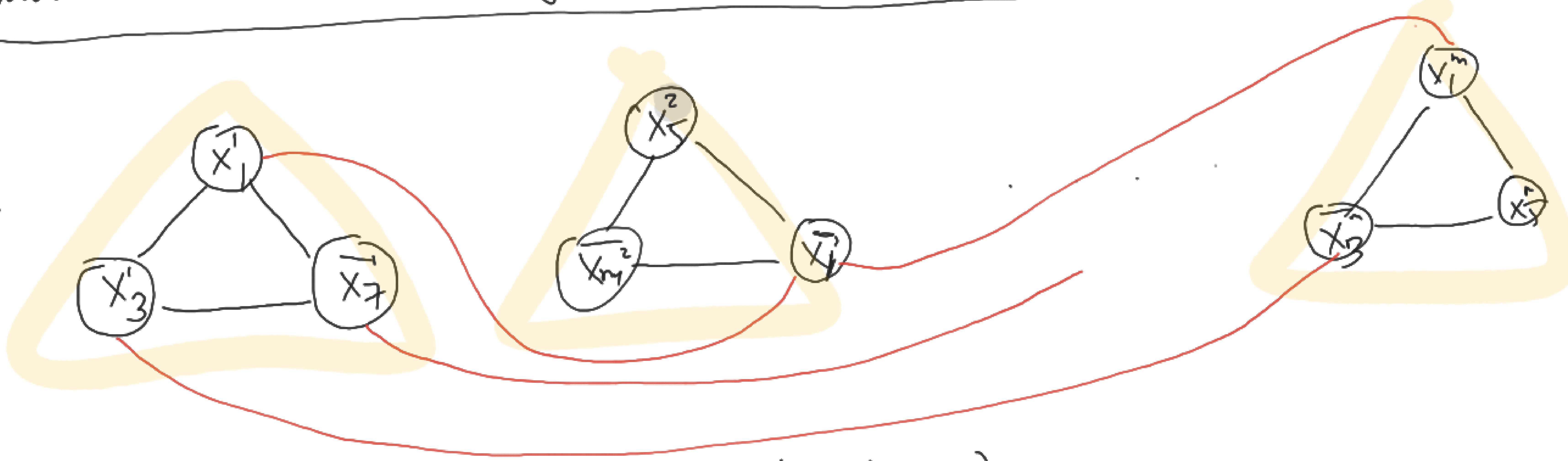
$\otimes$'

if $\underbrace{\text{IndSET} \in P}$ $\Rightarrow$ $\underbrace{\text{3SAT} \in P}$

3SAT: $\underbrace{(x_1 \vee x_3 \vee \bar{x}_7)}_{\text{clauses}} \wedge (x_5 \vee \bar{x}_{n-1} \vee \bar{x}_1) \wedge \cdots \wedge (x_1 \vee \bar{x}_3 \vee x_5)$ formula

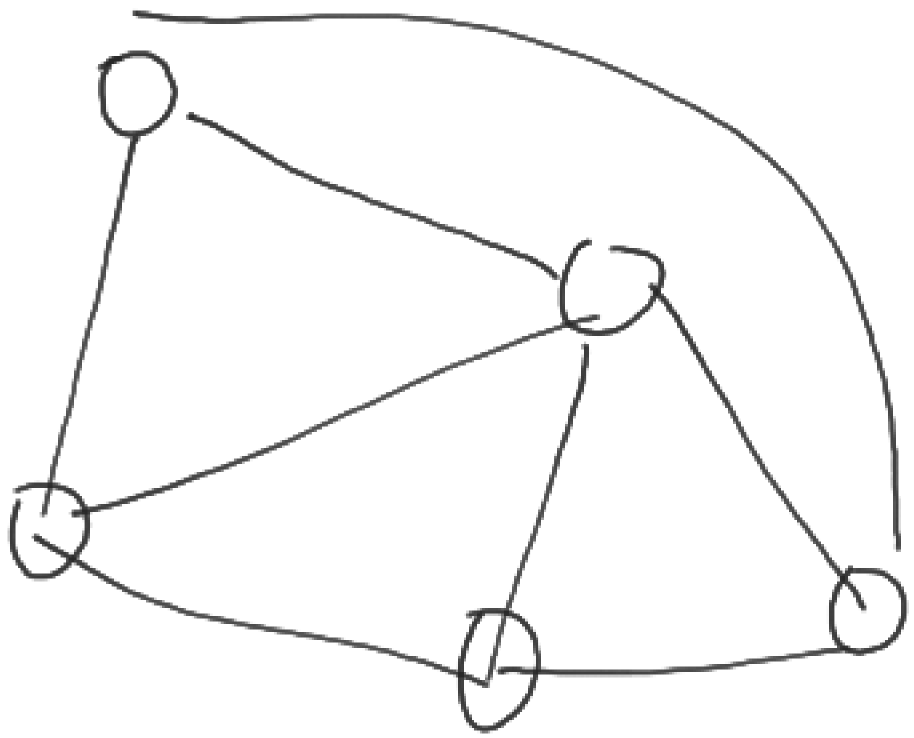
$\boxed{\text{Want to solve 3SAT using algo for IND SET}}$

$G =$



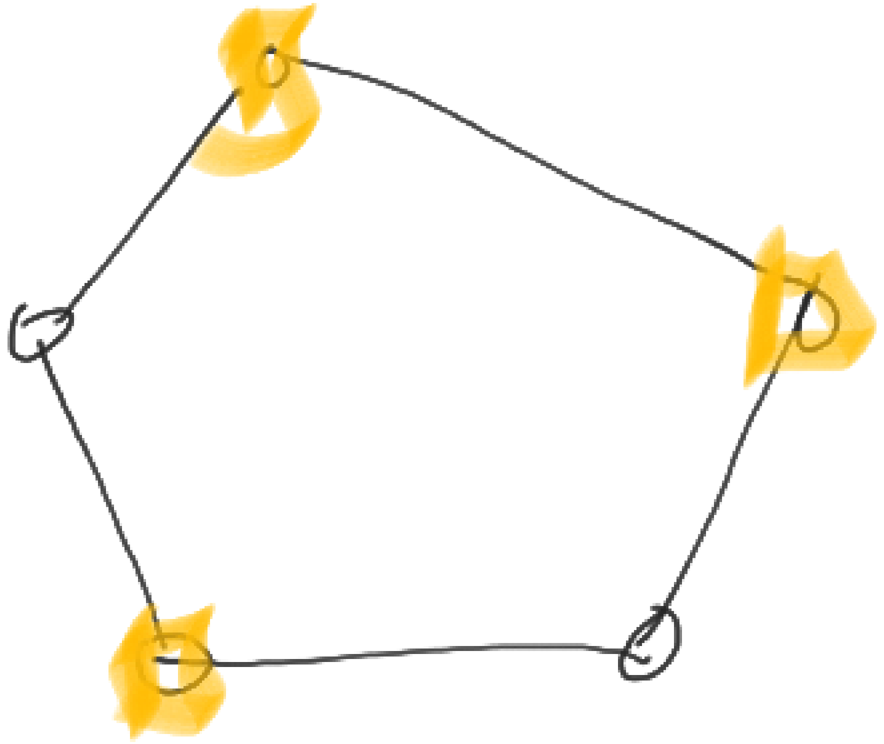
does $\exists$ an ind set of size $K = (\#\text{ clauses})$

Thm VERTEX COVER is NP-complete



$G=(V,E)$ C is a vertex cover if $\forall$ edge $(u,v) \in E$ C contains at least one of u or v.

(cover edges using vertices)

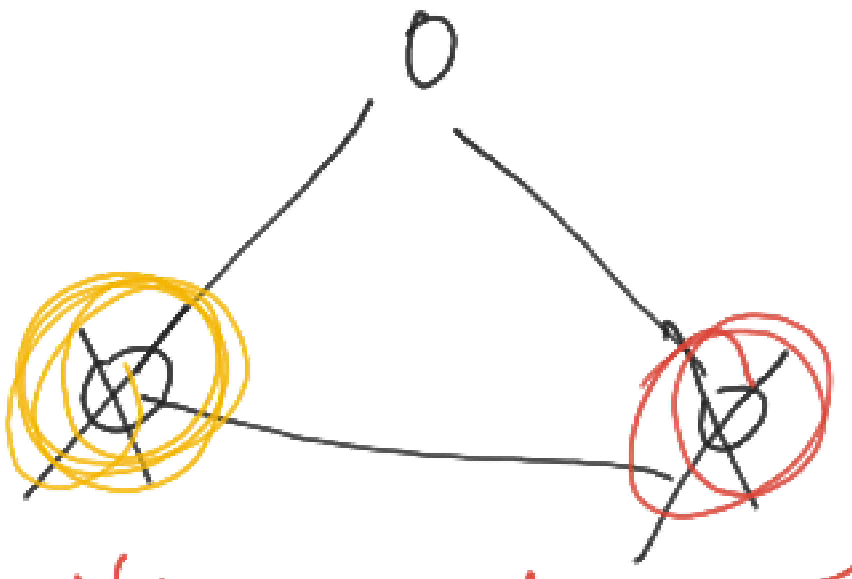


Problem Given G, K integer. does G contain a V.C of size $\leq K$

① VC is NP ✓

② if VC is P $\Rightarrow$ Ind Set is in P. ($\Rightarrow$ "which would be weird")

<u>Claim:</u> Graph G, S is an ind set $\Longleftrightarrow$ $V \setminus S$ is a vertex cover.



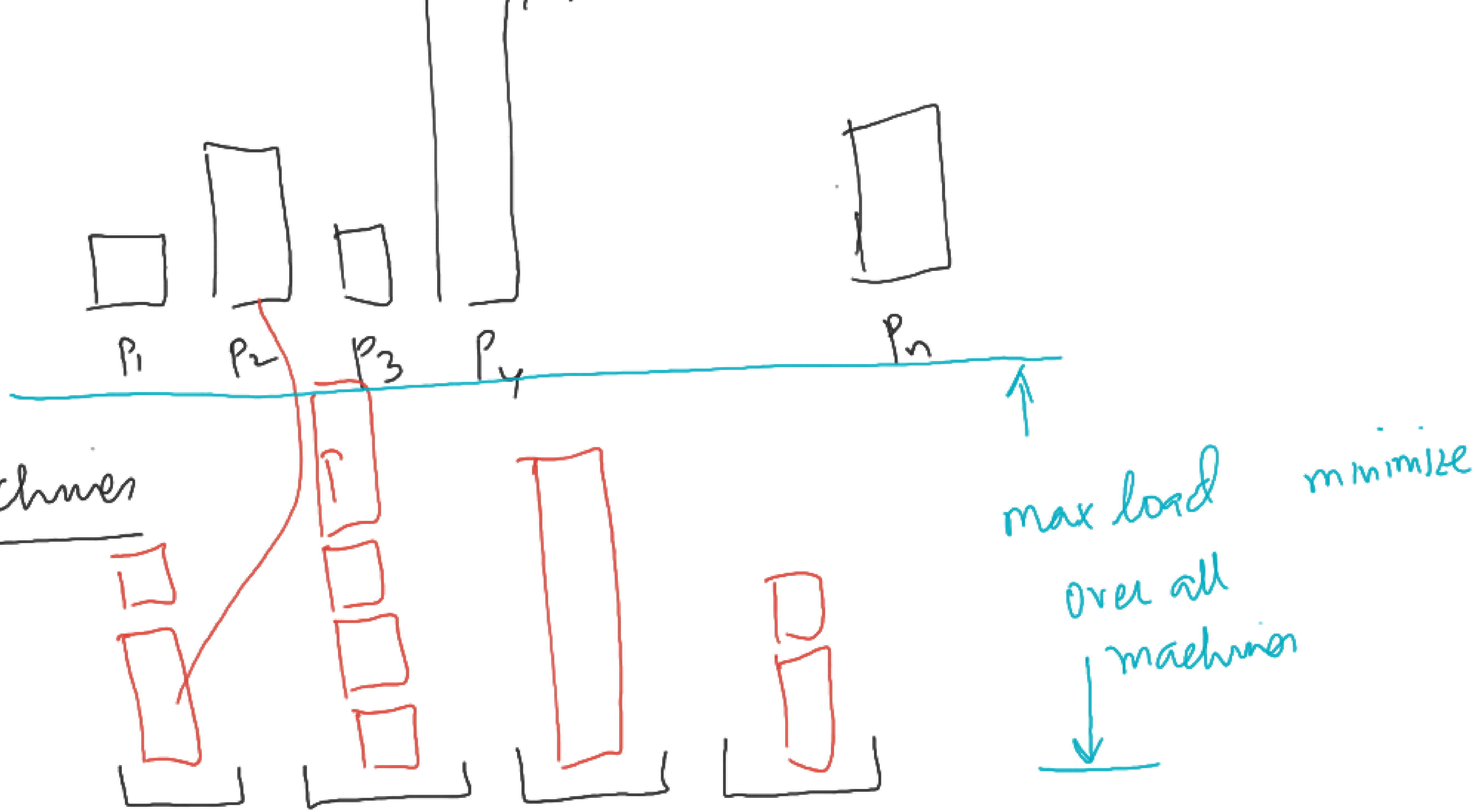
(G, K) is a YES instance of VC

$\Longleftrightarrow$ (G, n−K) is a YES instance of Ind SET

Load Balancing

Input: jobs "processing sizes"

1	2		n
p_1	p_2	$\dots$	p_n



m machines

Proc times 2 1 3 4 7 3 2 4 1

max load = "makespace"

~~11~~ ~~10~~ 9

$$OPT \geq \frac{\sum_i p_i}{m}$$

Q: Solve Makespan Minimization fast?

Thm 1: decision version of MM is NP complete

Thm 2: $\exists$ approx to MM.

if Partn $\in P \Rightarrow$ MM $\in P$

MM instance: $(m, p_1, p_2 \ldots p_n, K)$ does $\exists$ an assignment of jobs to machines with makespan $\leq K$

machines

sizes

Thm (1) MM $\in NP$. (hint = assignment)

(2) if MM $\in P \Rightarrow$ Partition $\in P$

$m = 2$

$p_i = a_i$

$K = \dfrac{\sum_{i \in L} a_i}{2}$

Partition: given integer numbers

$a_1, a_2 \ldots a_L \geq 0$

does $\exists$ a subset $S \subseteq \{1..L\}$

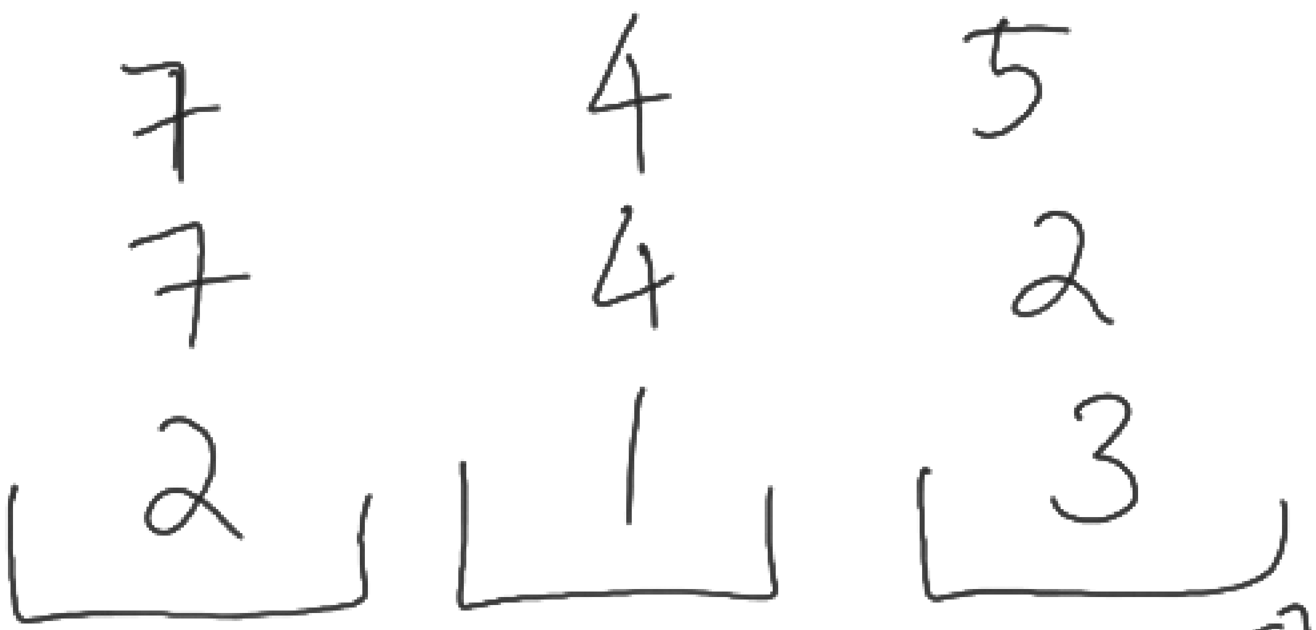
st $\sum_{i \in S} a_i = \sum_{i \notin S} a_i$

Thm: Partition is NP complete

Approximation Algo for Makespan Min

2 1 3 4 7 2 4 5 7 $\qquad m = 3$.

Greedy:
- sort jobs in decreasing order of size
- for $j = 1$ to n $\qquad$ // n jobs
  - put job j on least loaded machine so far



makespan of greedy $= 16$

$\left\lceil \frac{35}{3} \right\rceil \leq OPT \leq 14$

Fact: Makespan(Greedy) $\leq$ $2 \cdot \text{OPT}$ $\leftarrow$ Approx. Guarantee

worst case bound on the performance of a heuristic

Pf:

Greedy = last job on most loaded m/c + rest

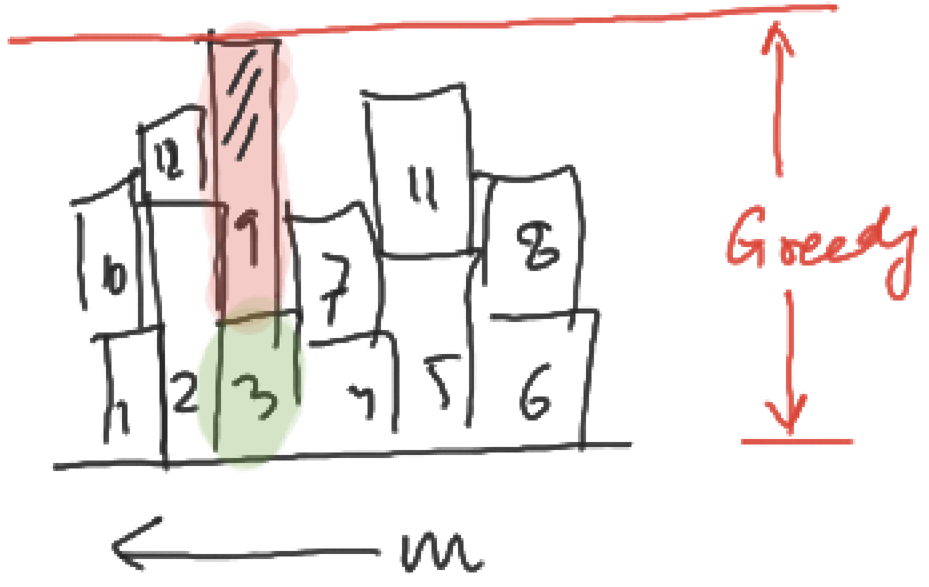
$$= P_{last} + (\text{rest})$$

$$\leq P_{max} + \frac{\sum_j P_j}{m}$$

$$\leq \text{OPT} + \text{OPT}$$

$$= 2\,\text{OPT}$$

☺



$$\frac{P_1 + P_2 + \cdots P_8}{m}$$

$$\leq \frac{\sum_j P_j}{m}$$

$$\boxed{P_{max} = \max_j P_j}$$

Thm: Greedy $\leq \left(2 - \frac{1}{m}\right)$ OPT

thm: $\exists$ examples where $\uparrow$ is tight

Thm: Sorted Greedy $\leq (1.5)$ OPT (Easyish)

$\left(1.5 - \frac{1}{m}\right)$ OPT (Easyish)

$\leq \left(\frac{4}{3} - \cancel{\frac{1}{m}}\right)$ OPT (Tricky)