L26

4102 11.26.2013

stable matching

abhi

Stable Matching

Image credits: Julia Nikolaeva

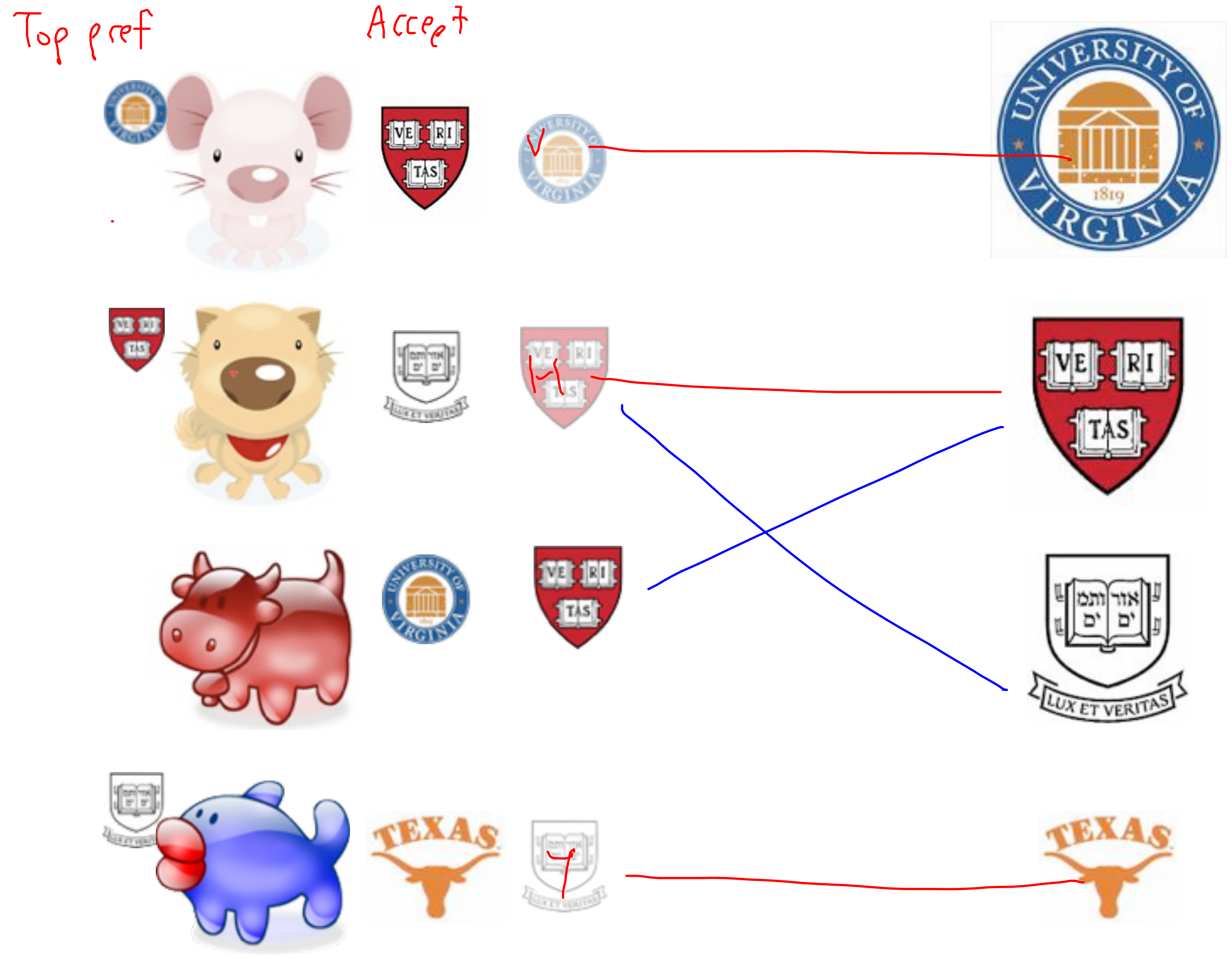


Image credits: Julia Nikolaeva

definition: matchings

- $M = \{m_1, \ldots m_n\}$
- $W = \{w_1, \ldots w_n\}$

$S = \{(m_i, w_j)\}$ set of pairs.

① matching if m_i, w_j only occur in one pair in S

② perfect if $|S| = n$.

definition: matchings

$$M = \{m_1, \ldots, m_n\}$$
$$W = \{w_1, \ldots, w_n\}$$

$$S = \{(m_{i_1}, w_{j_1}), \ldots, (m_{i_k}, w_{i_k})\}$$

Each m_i appears only one in a pairing.
A matching is perfect if every m_i appears.

definition: preferences

$$M = \{m_1, \ldots, m_n\}$$

each m_i has a preference list on the set W

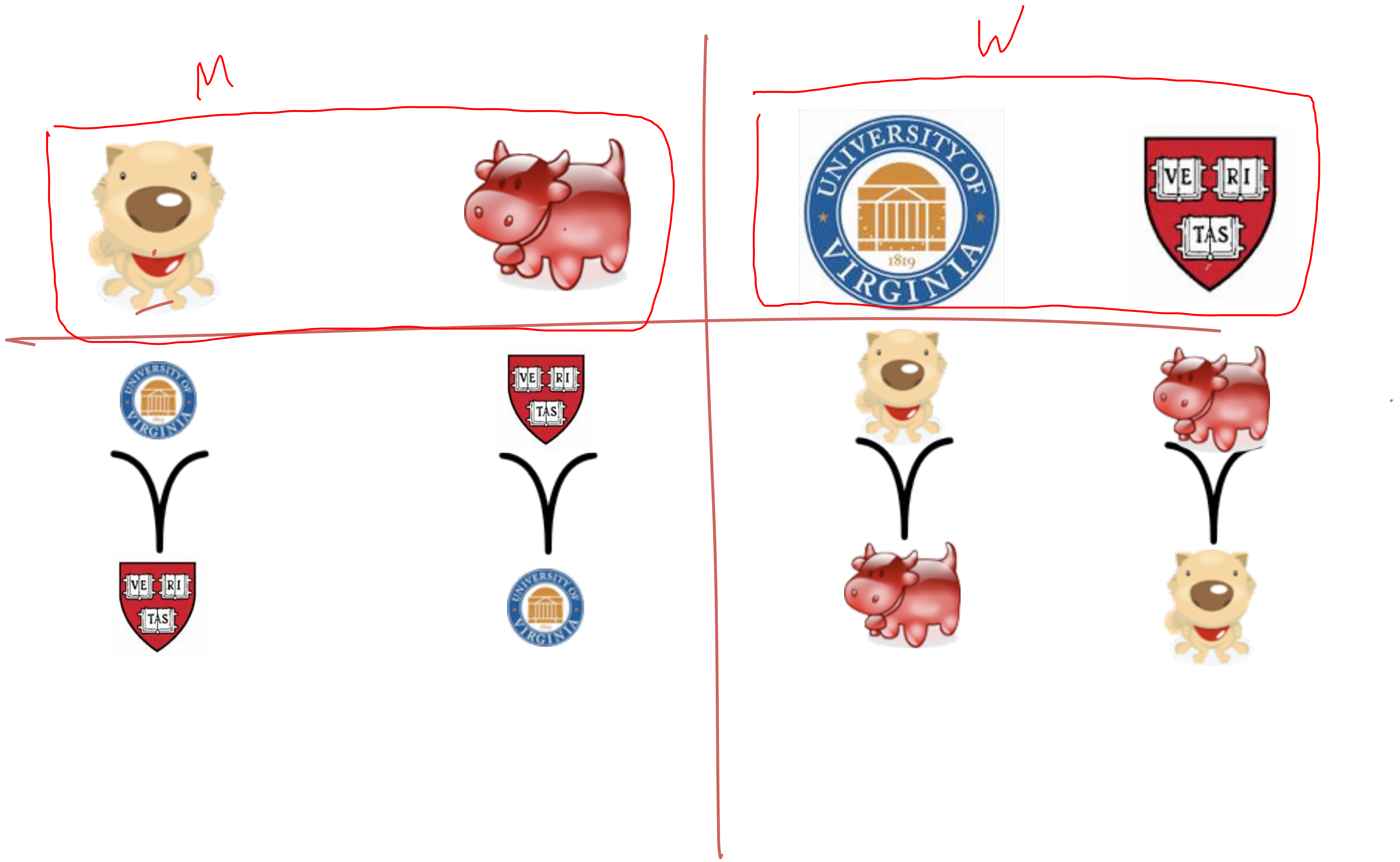
" $w_1 <_{m_i} w_2$ " : m_i prefers w_2 to w_1

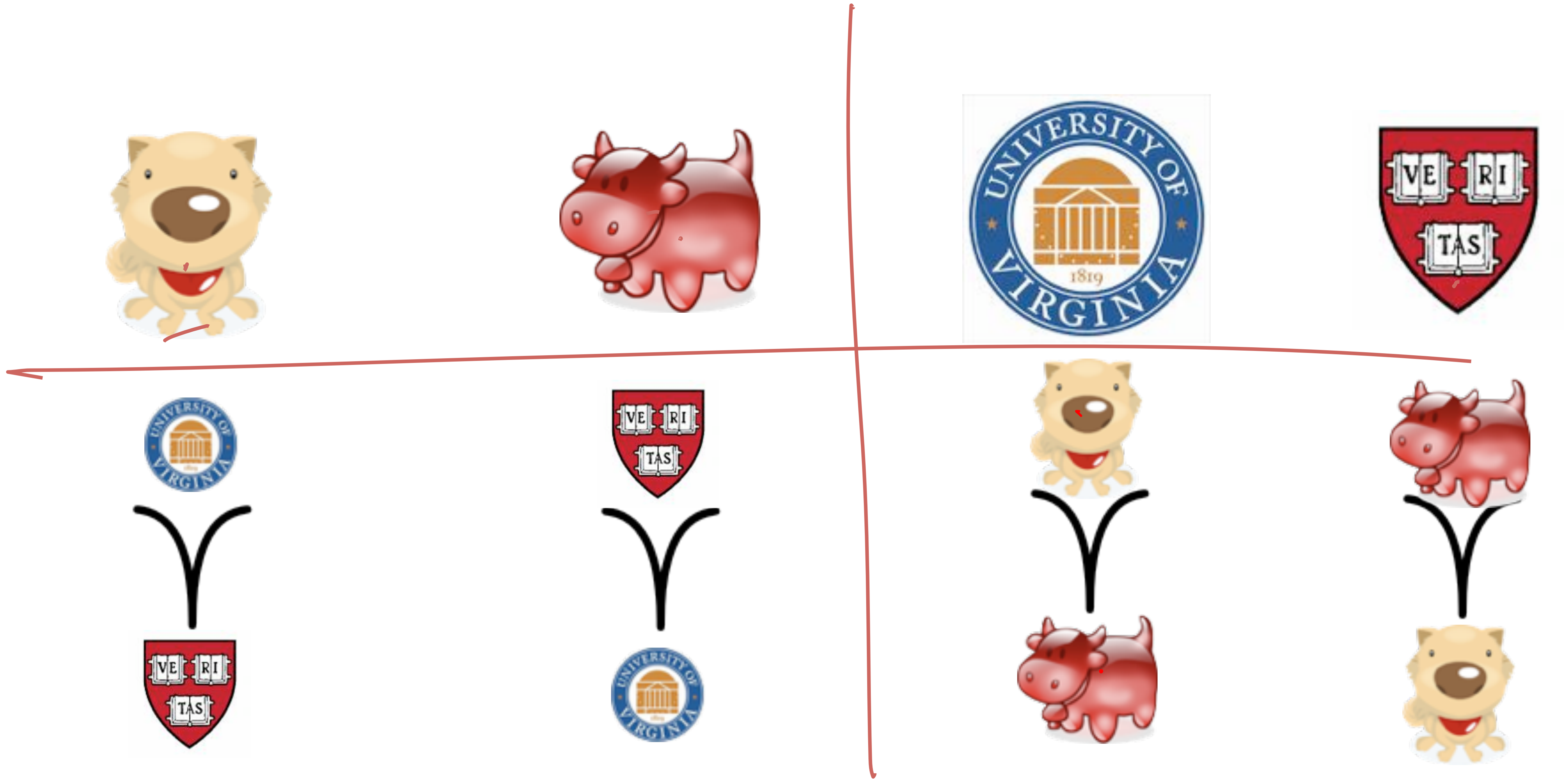
example: preferences

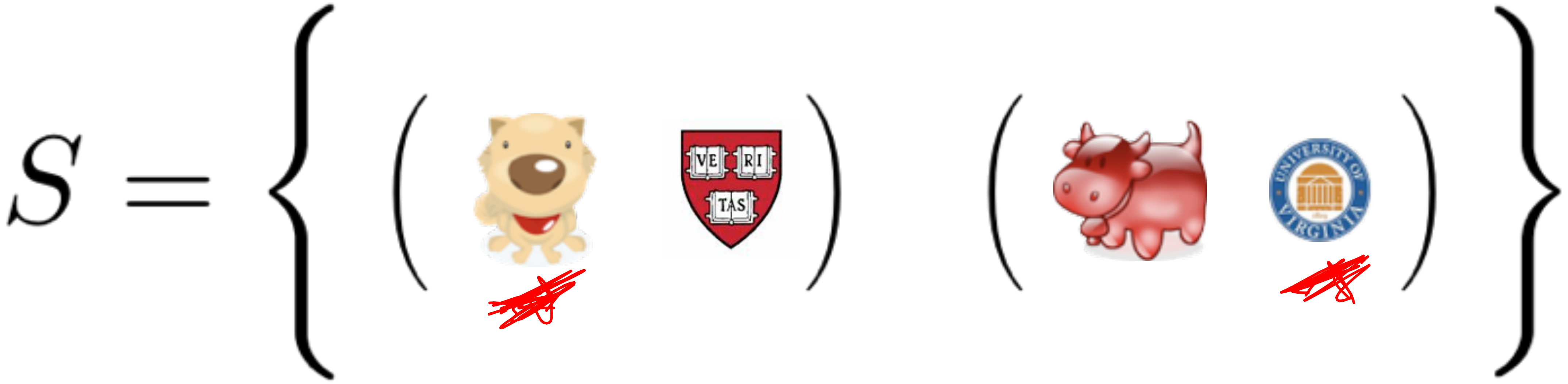
$$M = \{m_1, \ldots, m_n\}$$

m_i has a preference relation on the set W $\prec_{m_i}$

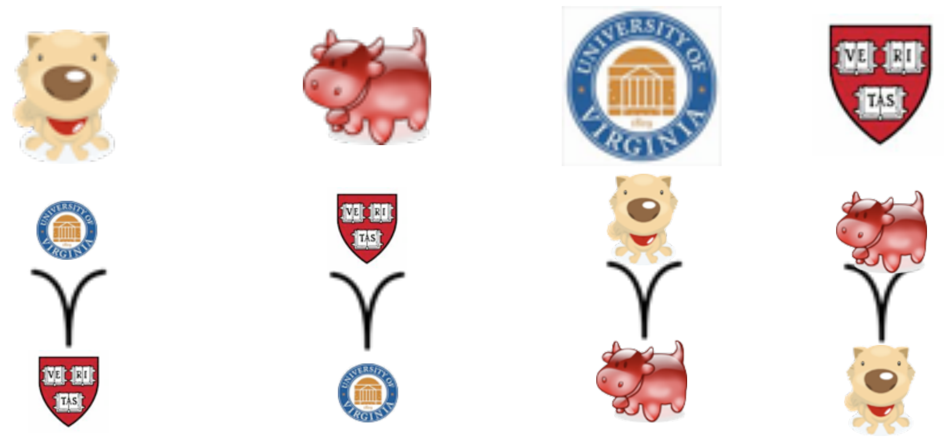
$$w_1 \prec_{m_i} w_4 \prec_{m_i} w_2 \prec_{m_i} w_8 \cdots w_n$$


M
W

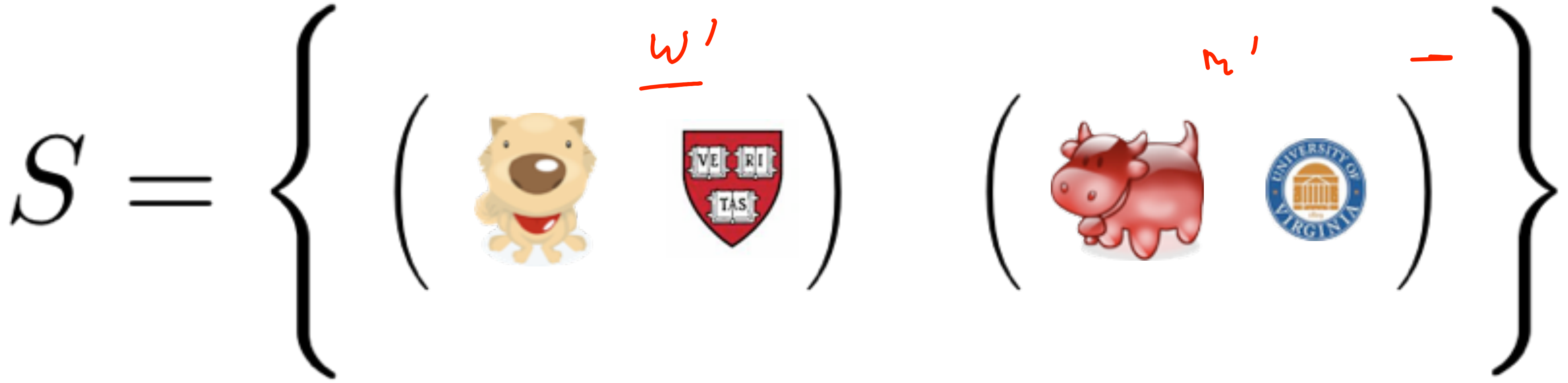




(D, V)   (C, H)



def: instability



S is unstable if $\exists\ (m^*, w^*) \notin S$, $(m^*, w'), (m', w^*) \in S$

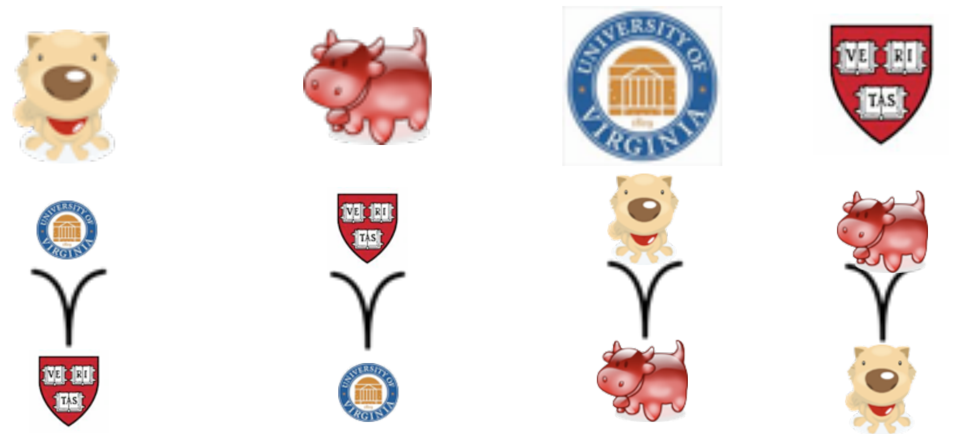
and

① m^* prefers w^* to w'

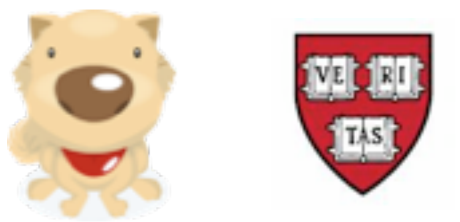
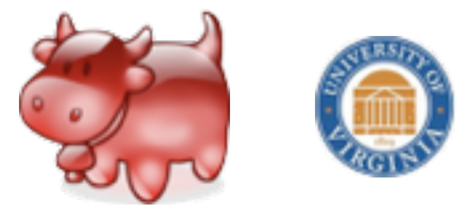
② w^* prefers m^* to m'

S is a stable matching if there are no such triples.

def: instability



w' m'

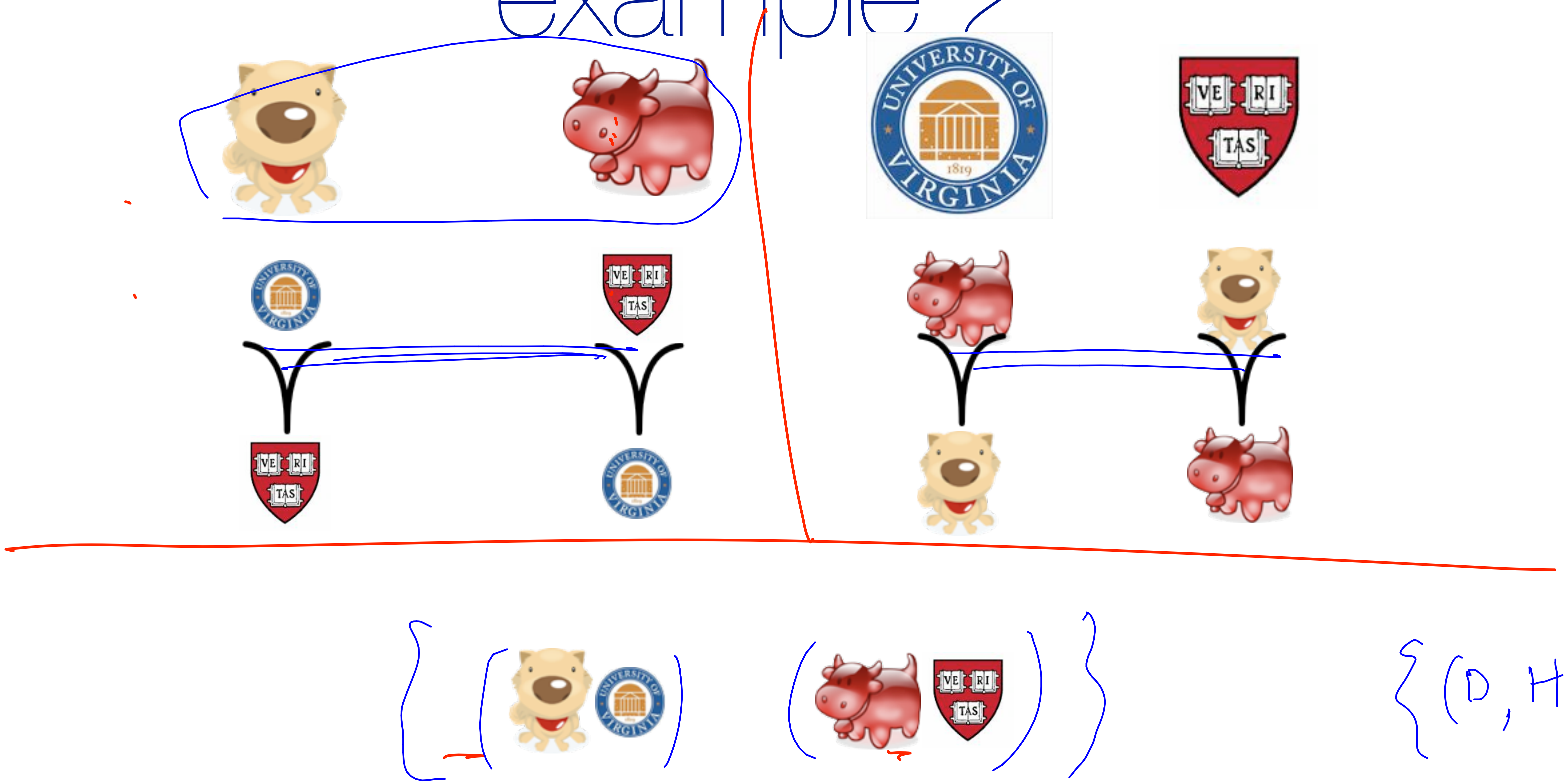
$$S = \Big\{ \Big($$  $$\Big) \quad \Big($$  $$\Big) \Big\}$$

$\Big($  $\Big)$ $\quad (m^*, w^*) \notin S$

$$w' \prec_{m^*} w^*$$

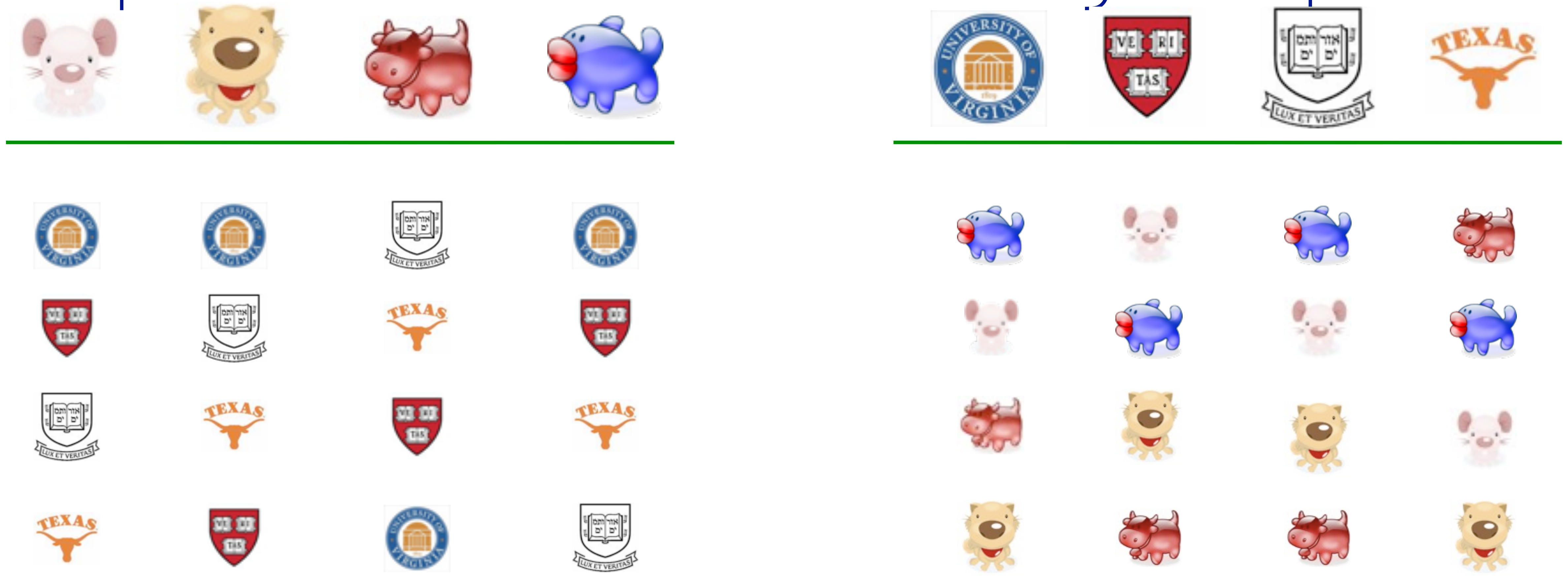
$$m' \prec_{w^*} m^*$$

example 2



$\{(D, H) \quad (C, V)\}$

prove: for every input



there exists a stable matching.

proposal algorithm

Start with an empty matching S

While $\exists$ an unmatched m who has not exhausted his preference list

Let w be the first ♀ on m's list who he has not asked

If w is unpaired, PAIR(m, w)

If $(m', w) \in S$ and w prefers m to m'

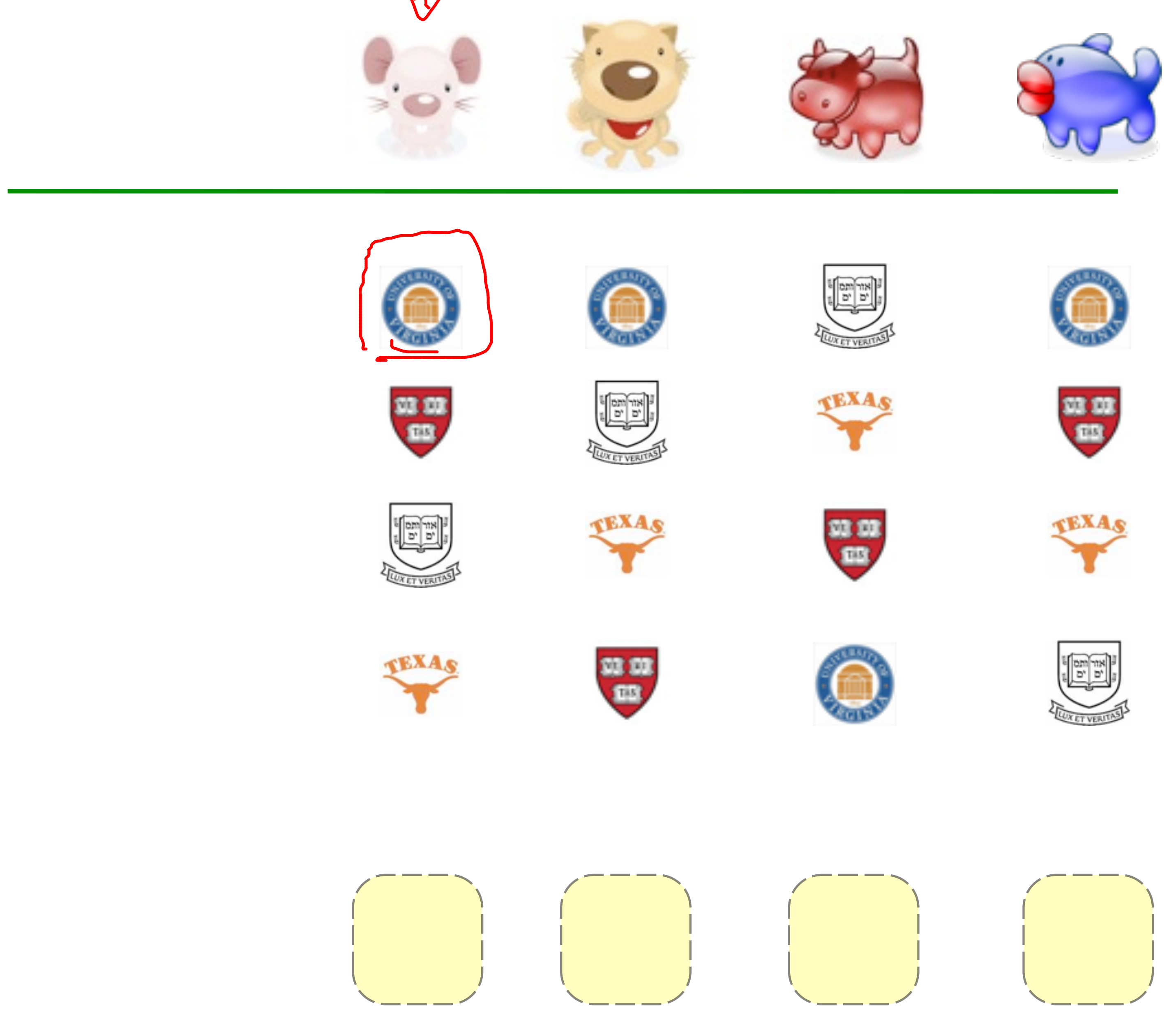
Breakup (m', w)

PAIR (m, w)

StableMatch($M, W, \prec_m, \prec_w$)

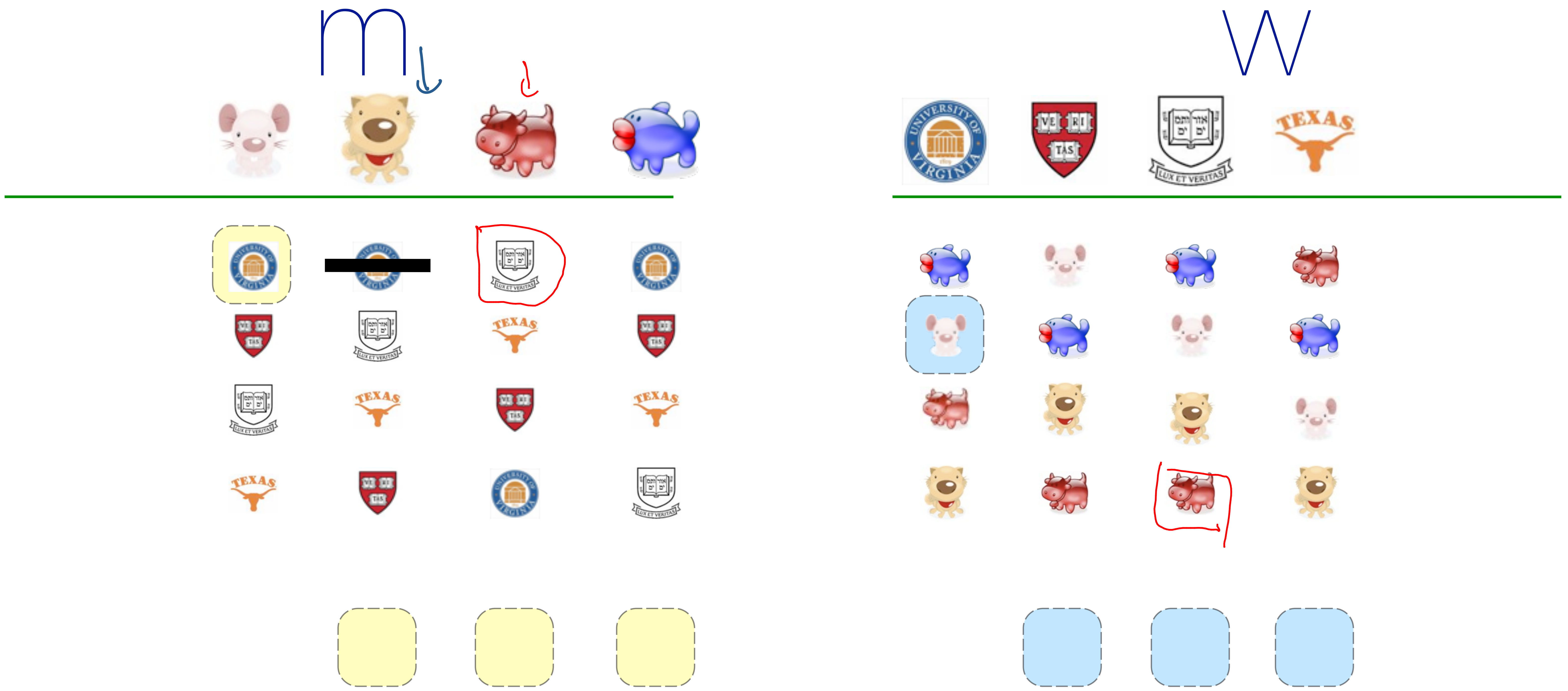
Initialize all m, w to be FREE
while $\exists$FREE(m) and hasn't proposed to all W
    do Pick such an m
        Let $w \in W$ be highest-ranked to whom m has not yet proposed
        if FREE(w)
            then Make a new pair (m, w)
        elseif (m', w) is paired and $m' \prec_w m$
            do Break pair (m', w) and make m' free
                Make pair (m, w)
return Set of pairs

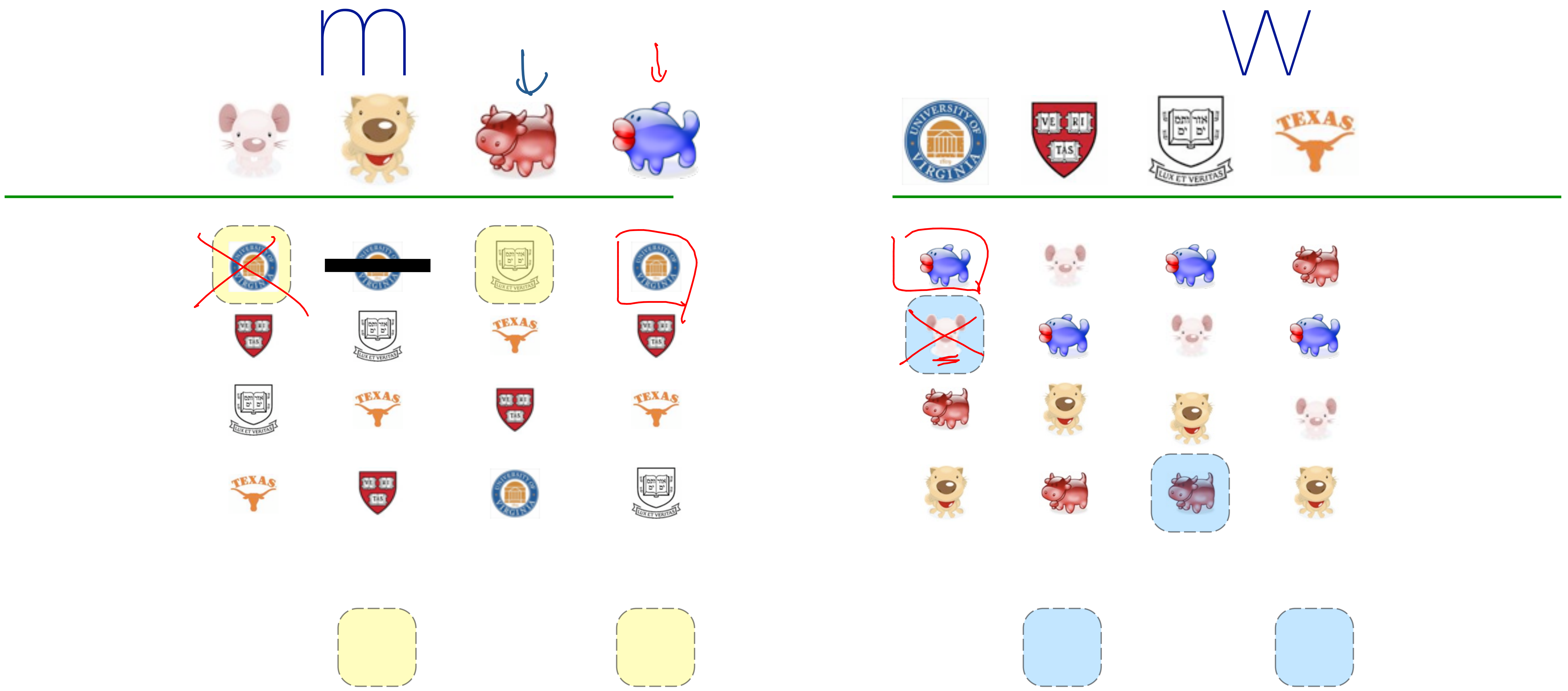
m



w

1st step


m
W


m
W


m
w


m
w


m
w


m
W

proposal algorithm ends

each m only asks each w once.

$|W| = n.$

$\Rightarrow \Theta(n^2)$ iterations of the loop.

proposal algorithm ends

$$O(n^2) \text{ steps}$$

each m proposes at most once to each w.

each m proposes at most n times.

size of M is n.

output is a matching

(1) EACH m only paired w/ one w.

(2) Why is each w paired w/ only 1 m ??

every time w is paired, she is single.

output is perfect

If $\exists$ an unmatched m,

some q has not been asked.

output is perfect

if $\exists m$ who is free, then $\exists w$ who has not been asked

output is stable $= S$

Spse not. That means $\exists (m^*, w^*) \notin S$ and (m^*, w'), $(m', w^*) \in S$
such that m^* prefers w^* to w' & w^* prefers m^* to m'.

$\to$ Consider the moment when (m^*, w') are paired in the execution.

m^* was single. m^* must have already asked w^*.

w^* must have rejected m^* in favor of $\hat{m}$.

$\Rightarrow \quad m^* <_{w^*} \hat{m}$

$\Rightarrow$ either $m' = \hat{m}$ or $\hat{m} <_{w^*} m'$

$\Rightarrow \quad m^* \leq_{w^*} m'$ which contradicts

(GALE-SHAPLEY)

output is stable

spse not. $\exists (m^*, w), (m, w^*) \in S \quad w \prec_{m^*} w^* \quad m \prec_{w^*} m^*$

output is stable

spse not. $\exists(m^*, w), (m, w^*) \in S$ $\quad w \prec_{m^*} w^* \quad m \prec_{w^*} m^*$

m* last proposal was to w

but $w \prec_{m^*} w^*$ and so m* must have already asked w*

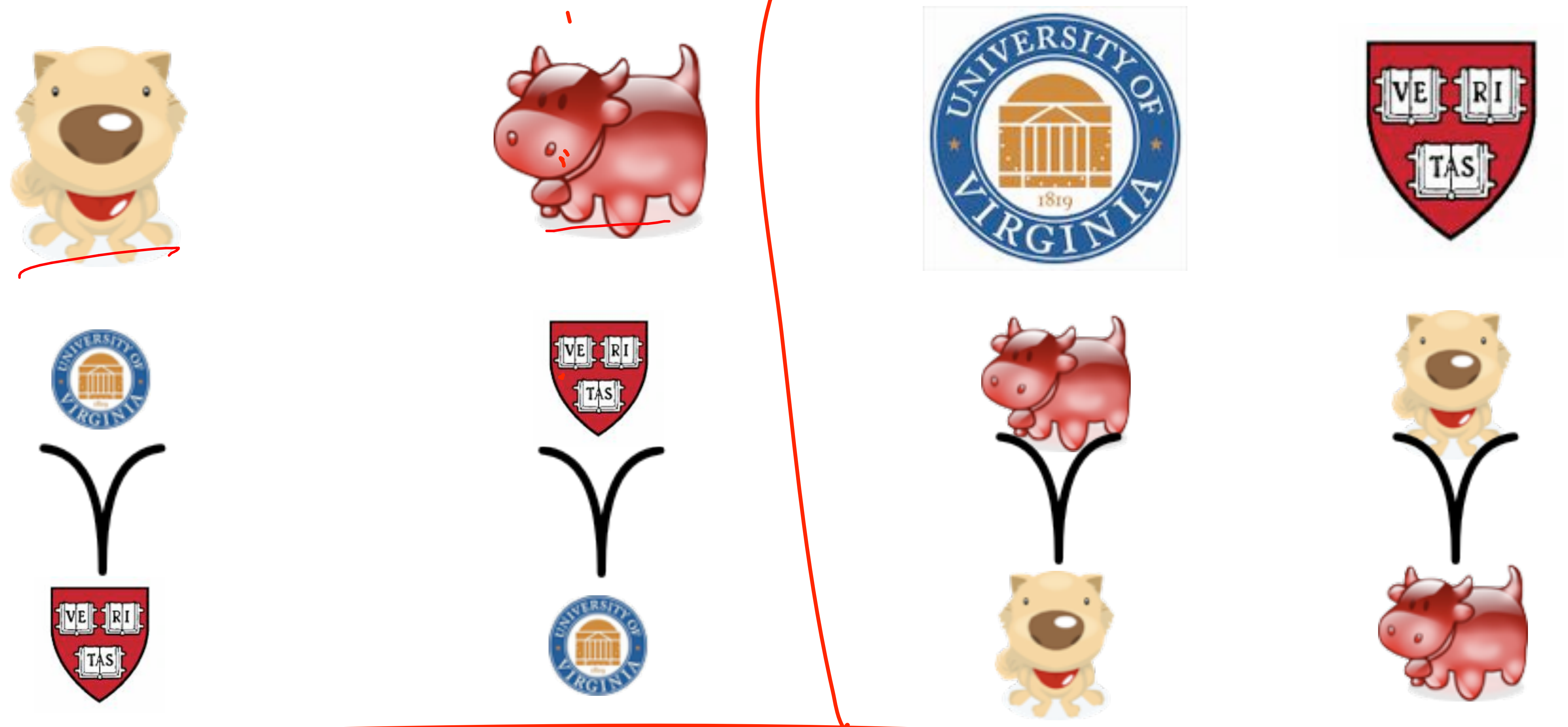
and must have been rejected by $m^* \prec_{w^*} m'$

then either $m' \prec_{w^*} m$ or m'=m

which contradicts assumption $m \prec_{w^*} m^*$

Proposer wins

M

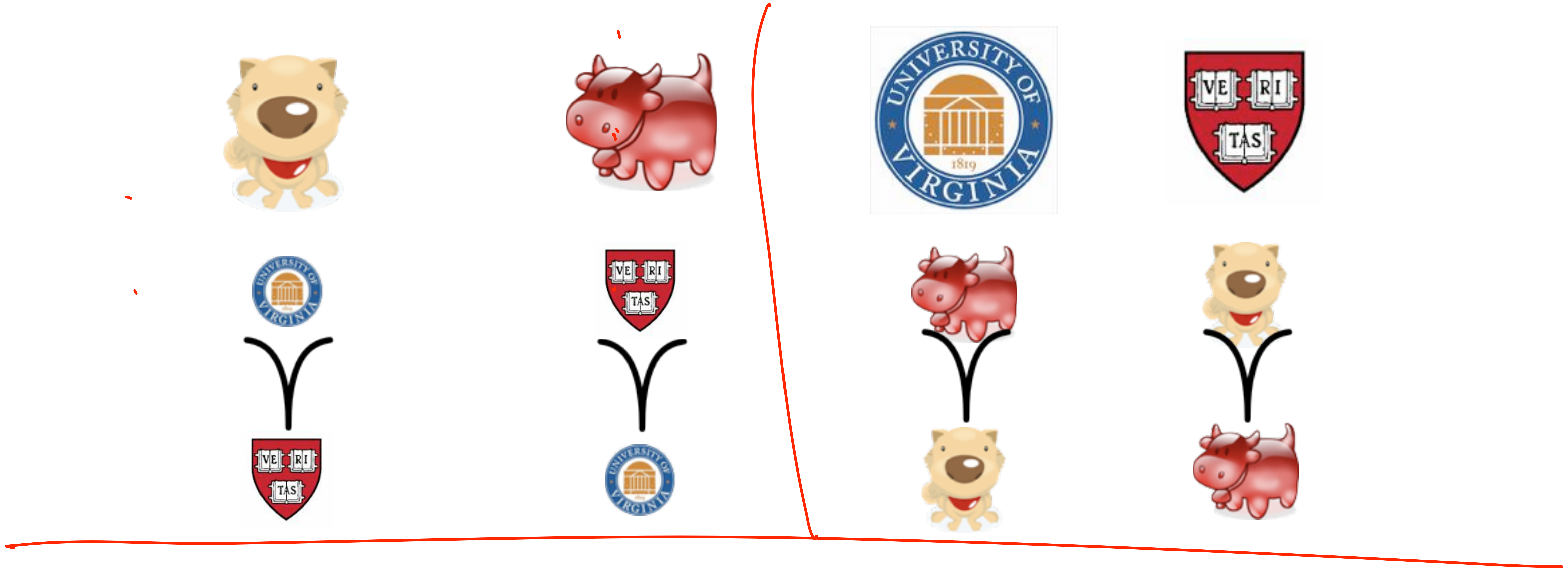


(D, U), (C, H)

∃ propose

(C, U) (D, H)

Proposer wins



Remarkable theorem

w is valid for m: $\exists$ some stable matching S s.t. $(m, w) \in S$

best(m): w s.t. (m, w) is VALID ~~any~~ every $w^* \underset{m}{>} w$ is not VALID.

$$S^* : \{ (m, best(m)) \}_{m \in M}$$

Thm: GS returns S^*. (every execution of it).

GS is man-optimal.

Proof: Consider some execution E of GS that returns $S \neq S^*$.

$\Rightarrow$ some m is not matched with their best VALID match.

$\Rightarrow$ some w must have rejected a VALID m.

$\Rightarrow$ Consider the first time some w rejects a VALID m. — ask + reject / breakup.

@ this point, (m', w) have been matched

But w is VALID for m, so $\exists\ (m, w) \in S'$.

who does m' match with in S' ?? $(m', w') \in S'$.

→ Since this is the first rejection in E, then w' could not have rejected m' in E at this point.

(To be fixed).

GS matching vs Wopt

MAN- optimal.

- The MAN- optimal matching is the worst-possible stable matching for W.

THE MATCH℠

NATIONAL RESIDENT MATCHING PROGRAM®

a new technique
for algorithm
design

MergeSort(n)

<base case>

MergeSort(n/2) <left half>

MergeSort(n/2) <right half>

Merge(left,right) <combine>

MergeSort(n)

<base case>

MergeSort(n/2) <left half>

MergeSort(n/2) <right half>

Merge(left,right) <combine>

$$T(n) = 2T(n/2) + O(n)$$

MergeSort(n) $<_{f(n)}$ 2MergeSort(n/2)

Typesetting

$$\mathrm{BEST}_n = \min \left\{ \begin{array}{l} \mathrm{BEST}_0 + S_{1,n}^2 \\ \mathrm{BEST}_1 + S_{2,n}^2 \\ \mathrm{BEST}_2 + S_{3,n}^2 \\ \quad\ldots \\ \mathrm{BEST}_{\ell-1} + S_{\ell,n}^2 \\ \\ \quad\ldots \\ \\ \mathrm{BEST}_{n-1} + S_{n,n}^2 \end{array} \right.$$

Typesetting

$$\mathrm{BEST}_n = \min \begin{cases} \mathrm{BEST}_0 + S^2_{1,n} \\ \mathrm{BEST}_1 + S^2_{2,n} \\ \mathrm{BEST}_2 + S^2_{3,n} \\ \quad \cdots \\ \mathrm{BEST}_{\ell-1} + S^2_{\ell,n} \\ \\ \quad \cdots \\ \mathrm{BEST}_{n-1} + S^2_{n,n} \end{cases}$$

solving BESTn can be reduced to solving n-1 BESTi problems and combining the answer in linear time.

HUFFMAN

Finding an optimal code for an X character alphabet

solved by

can be reduced to

Finding an optimal code for an X-1 character alphabet

WE HAVE BEEN SOLVING PROBLEM A
BY SOLVING SMALLER VERSIONS OF
PROBLEM A

GENERAL IDEA:

SOLVE PROBLEM A BY SOLVING PROBLEM B

REDUCTION

$$\text{PROBLEM}_{a} \leq_{f(n)} \text{PROBLEM}_{b}$$

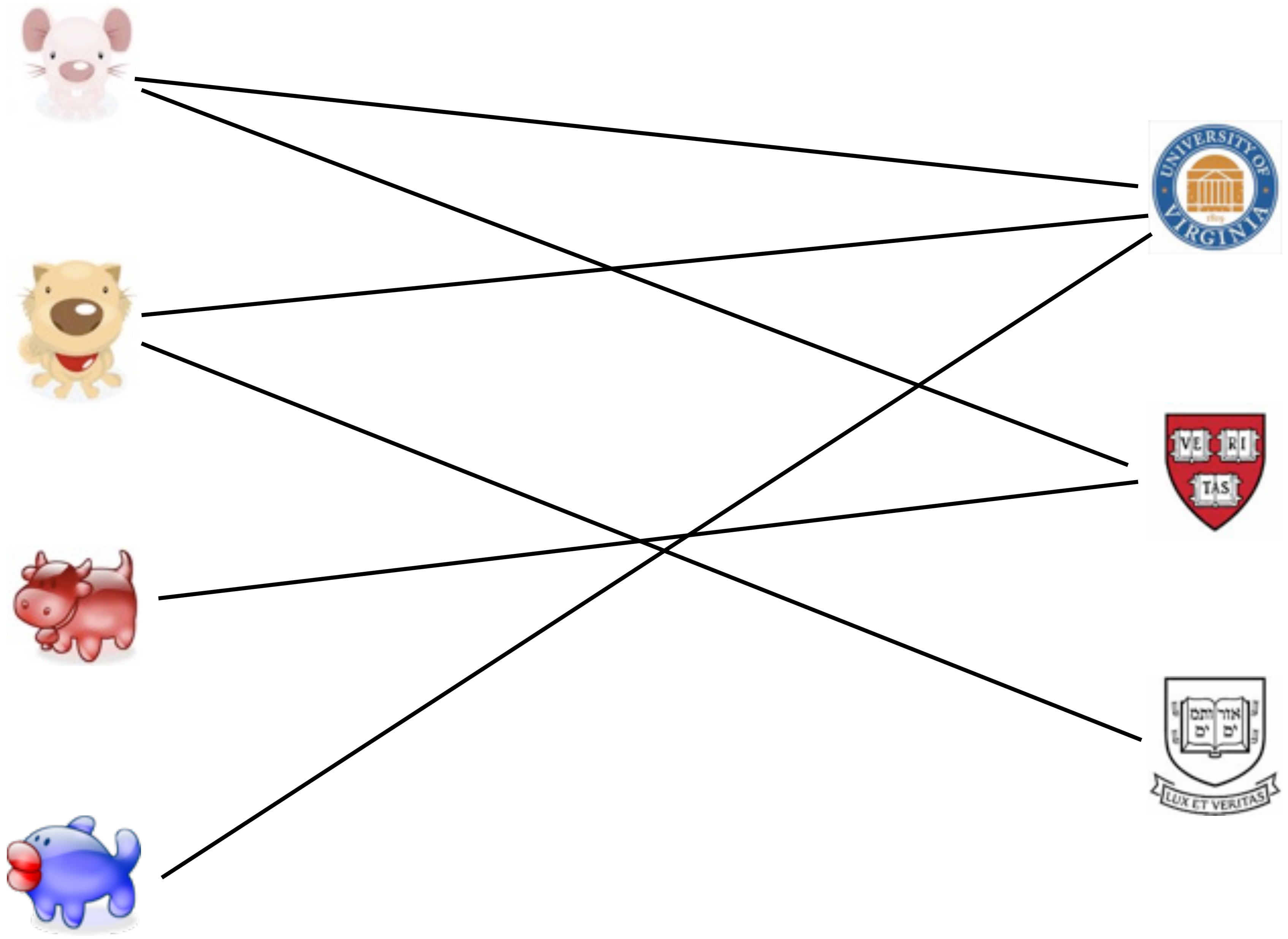
REDUCTION

$$\text{PROBLEM}_a \leq_{f(n)} \text{PROBLEM}_b$$

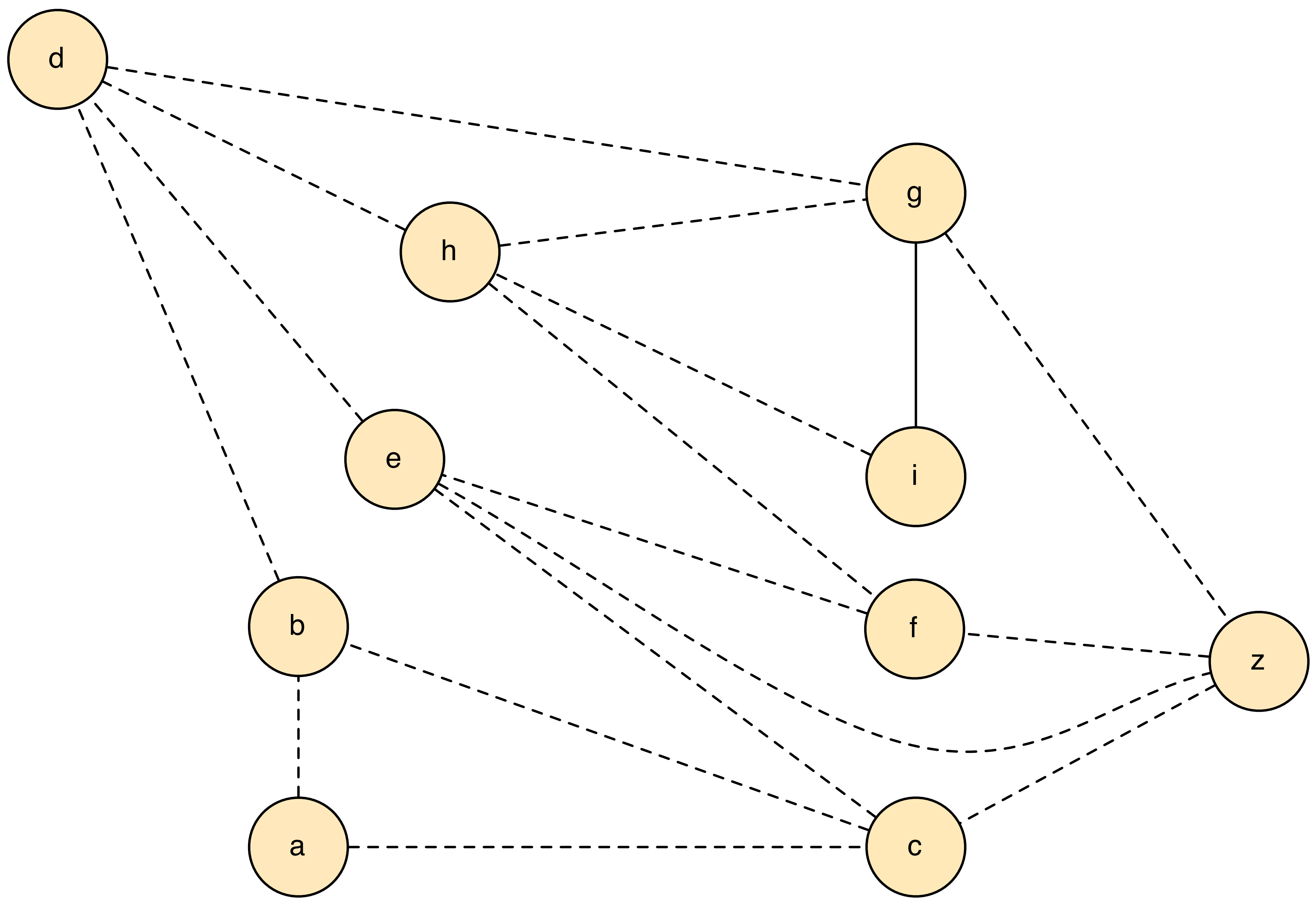
$\exists c, d$

$$T(\text{PROBLEM}_a(n)) \leq f(n) + cT(\text{PROBLEM}_b(dn))$$

MAXIMUM BIPARTITE MATCHING



EDGE-DISJOINT PATHS



$\text{MAXBIPARTITE} <_{e+v} \text{maxflow}$

$\text{maxedgedisj} <_{e+v} \text{maxflow}$

TRIPLET PROBLEM

given numbers $(x_1, \ldots, x_n)$

determine whether there is a triplet (x_i, x_j, x_k)

such that $x_i + x_j + x_k = 0$

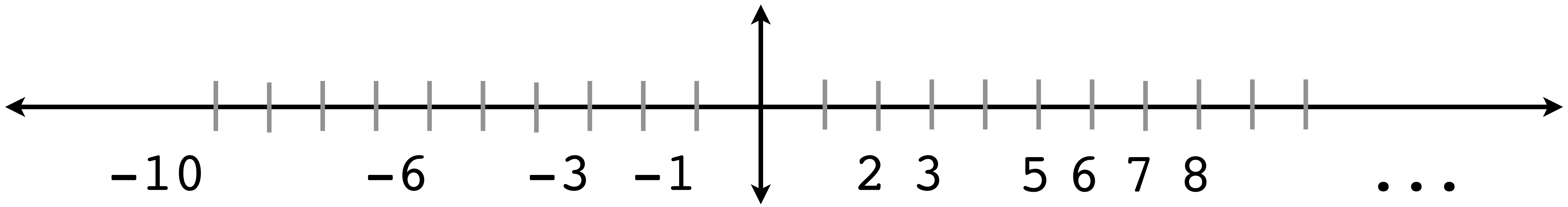
3,-6, 5 ,2,6,8,-1,12,7,-10,-3,14

EASY TO SOLVE IN

$$O(n^3)$$

EASY TO SOLVE IN $O(n^2)$

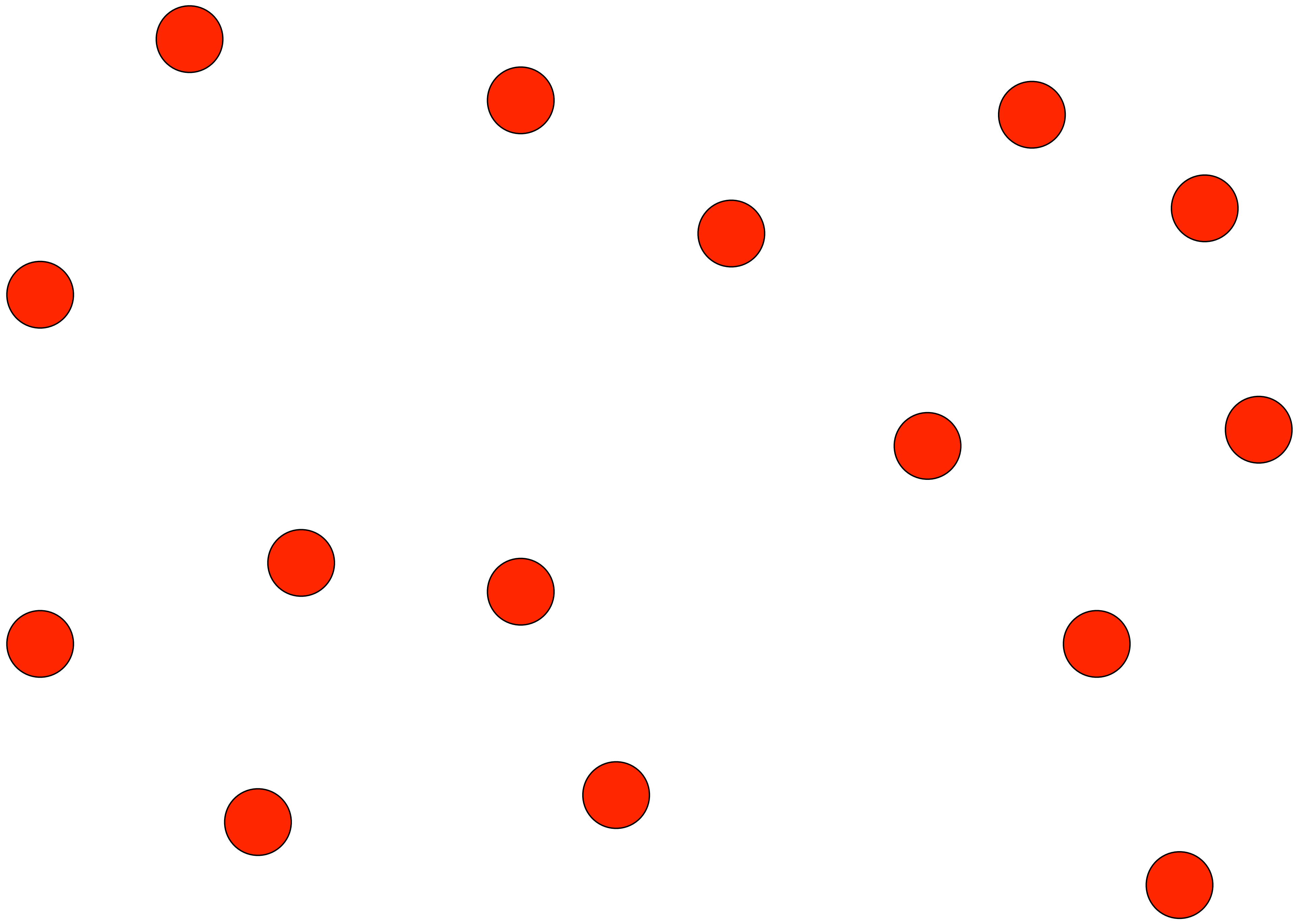
3,-6, 5 ,2,6,8,-1,12,7,-10,-3,14



COLINEARITY

given points in the plane $((x_1, y_1), \ldots, (x_n, y_n))$

determine whether any 3 are co-linear but not horizontal.

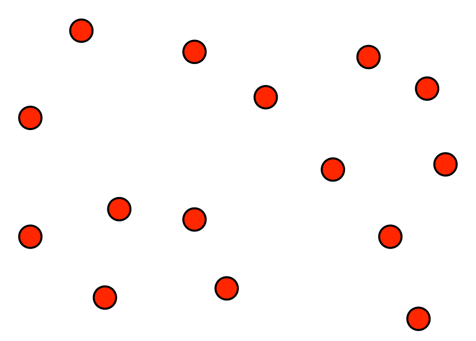


HOW CAN WE COMPARE 2 PROBLEMS?

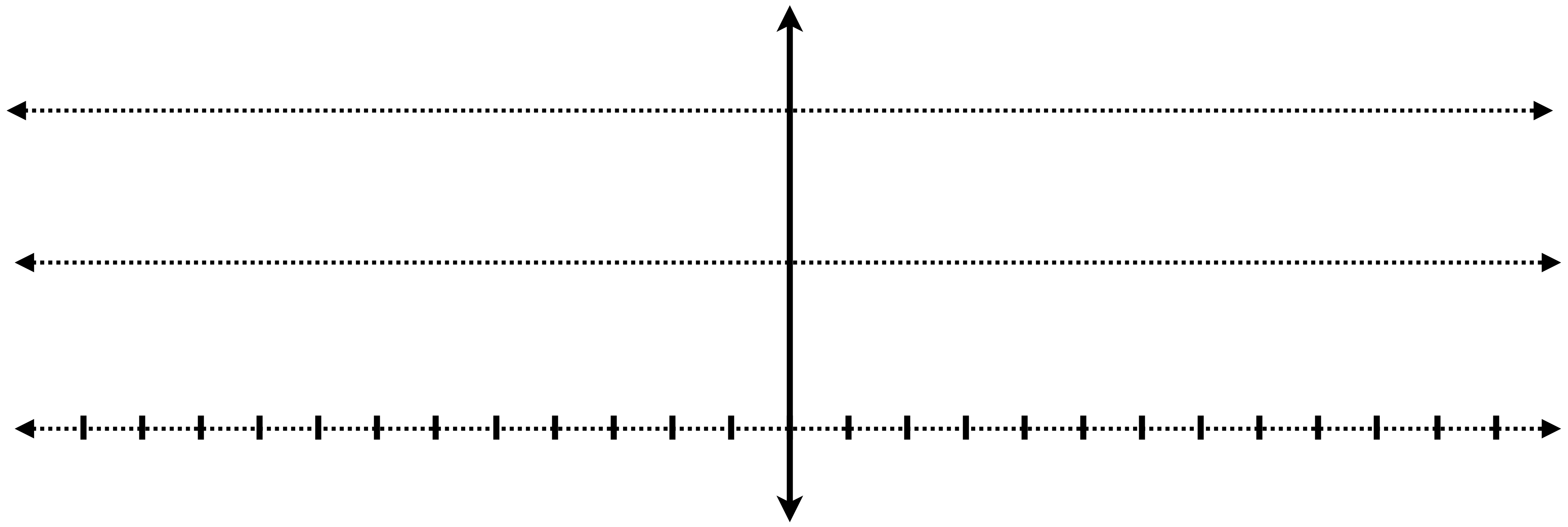
$$\text{PROBLEM}_a \leq_{f(n)} \text{PROBLEM}_b$$

$$T(\text{PROBLEM}_a(n)) \leq f(n) + cT(\text{PROBLEM}_b(dn))$$

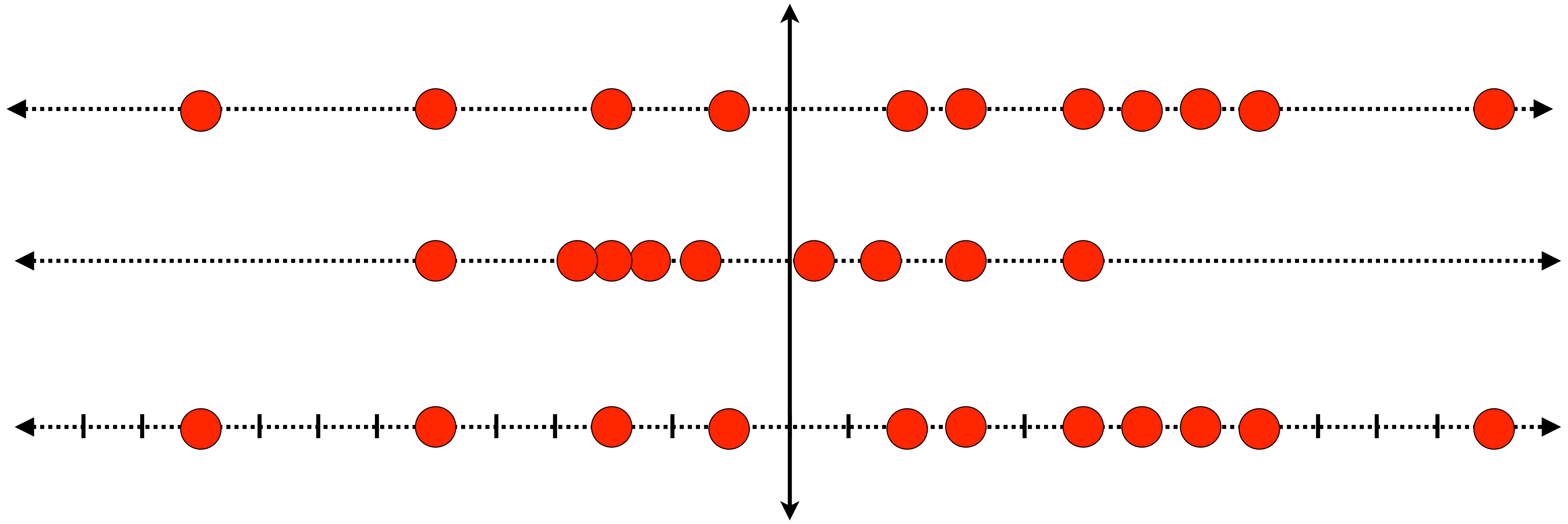
3,-6,5,2,6,8,-1,12,7,-10,-3,14



$T = \{\ 3, -6, 5, 2, 6, 8, -1, 12, 7, -10, -3, 14\ \}$



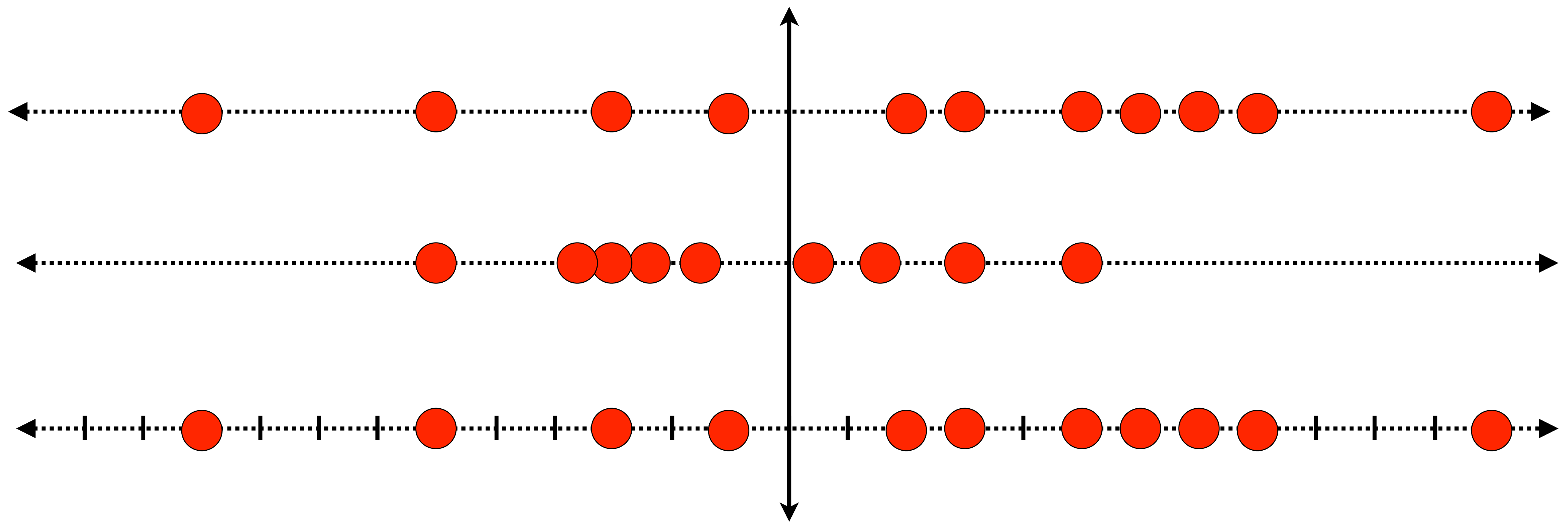
T= { 3,-6,5,2,6,8,-1,12,7,-10,-3,14 }



P=

T= { 3,-6,5,2,6,8,-1,12,7,-10,-3,14 }

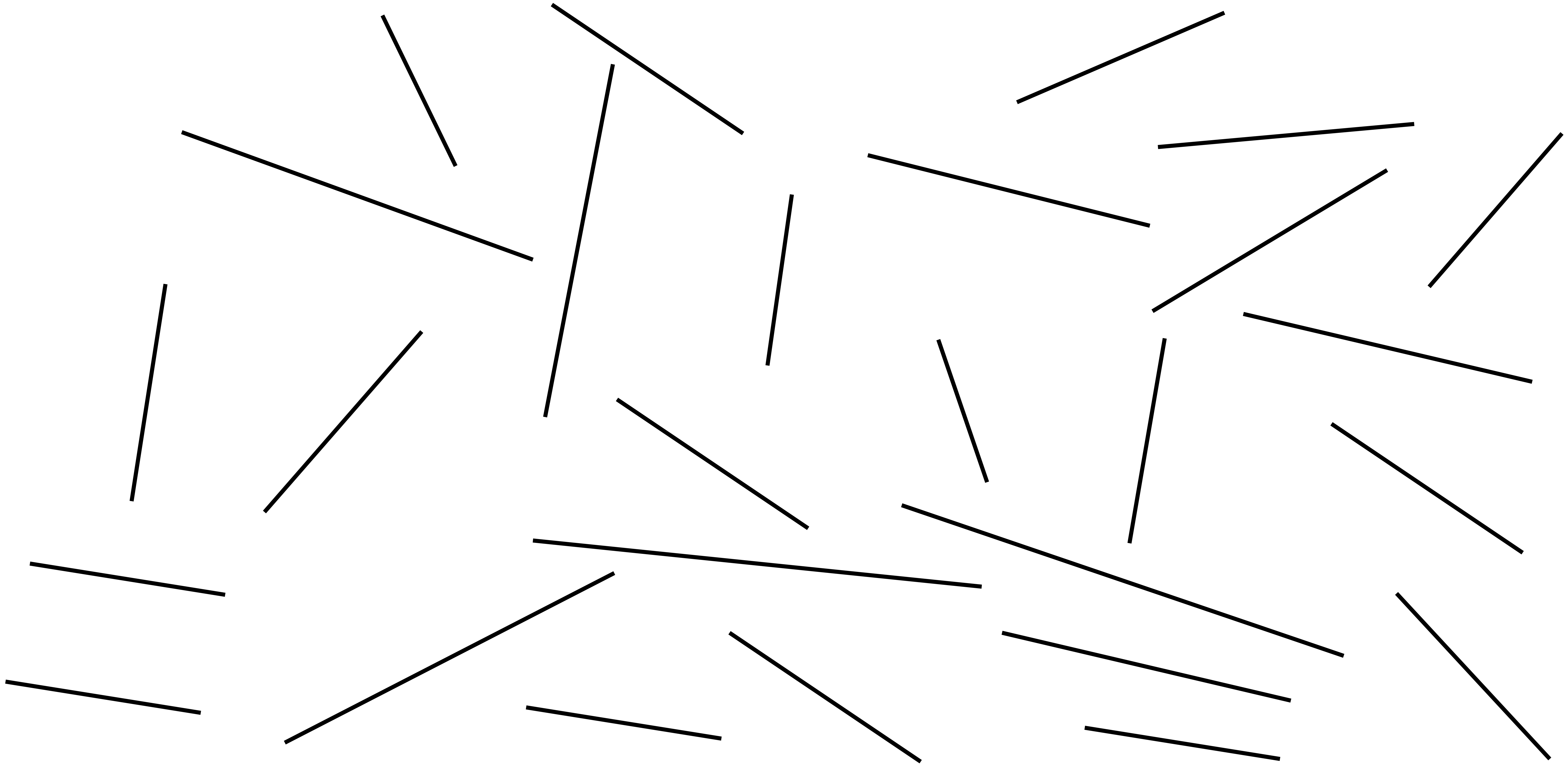
P=



T is a TRIPLET-set if and only if P is a COLINEAR set.

SEGMENT PARTITION

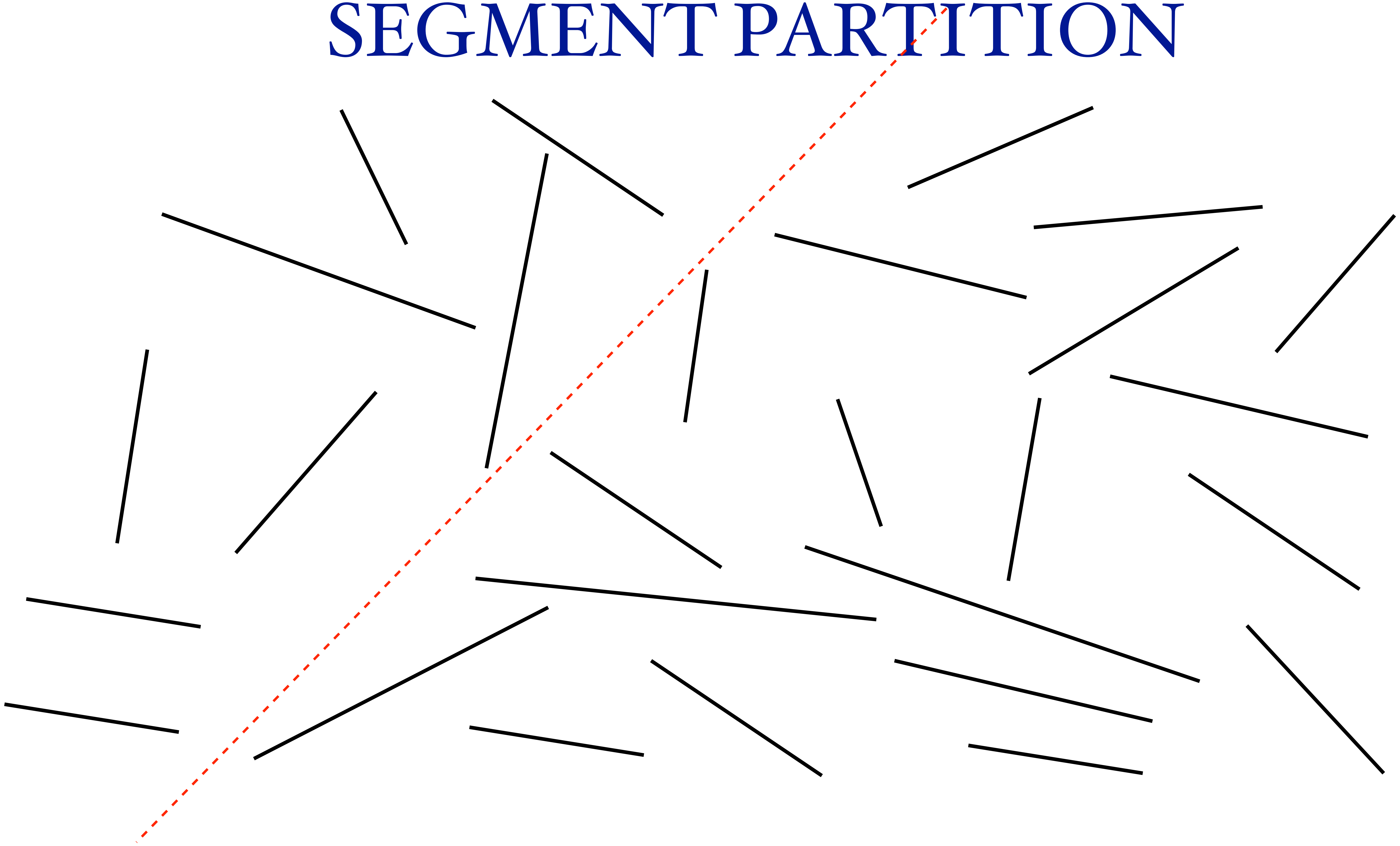
SEGMENT PARTITION



SEGMENT PARTITION

Problem: Given a set of line segments in the plane, determine if there exists a line that partitions the segments into two sets.

SEGMENT PARTITION



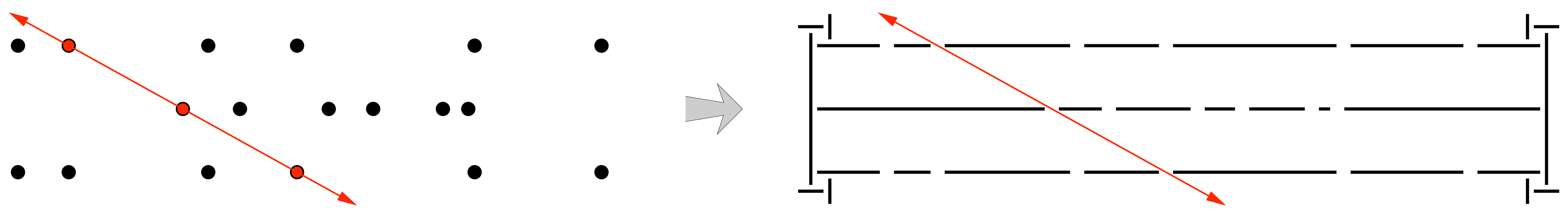


image: erickson

WHY DO WE CARE?

ANOTHER EXAMPLE

3SAT PROBLEM

input:

output: “

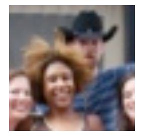
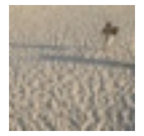
3SAT EXAMPLE

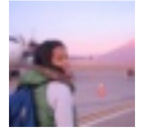
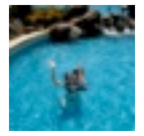
$$(x \vee y \vee z) \wedge (x \vee \overline{y} \vee y) \wedge (u \vee y \vee \overline{z}) \wedge (z \vee \overline{x} \vee u) \wedge (\overline{x} \vee \overline{y} \vee \overline{z})$$

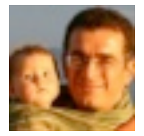
PARTY PROBLEM





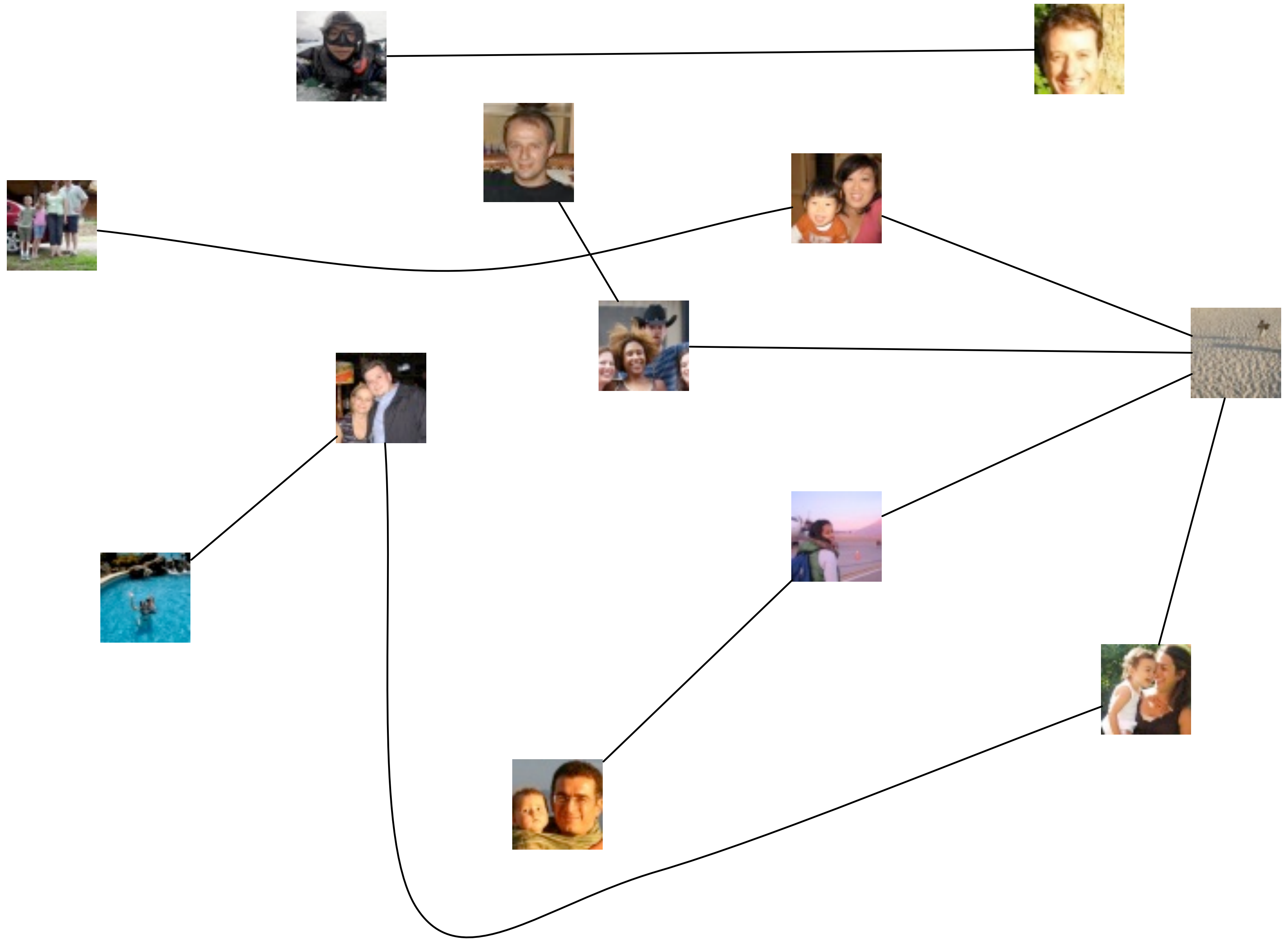


INDEPENDENT SET

INDEPENDENT SET

a set $S \subseteq V$ is an independent set if
no two nodes in S are joined by an edge.

EXAMPLE



GOAL: given a graph G,

$3\text{SAT} \leq_p \text{INDSET}$

$$(x \vee y \vee z) \wedge (x \vee \overline{y} \vee y) \wedge (u \vee y \vee \overline{z}) \wedge (z \vee \overline{x} \vee u) \wedge (\overline{x} \vee \overline{y} \vee \overline{z})$$

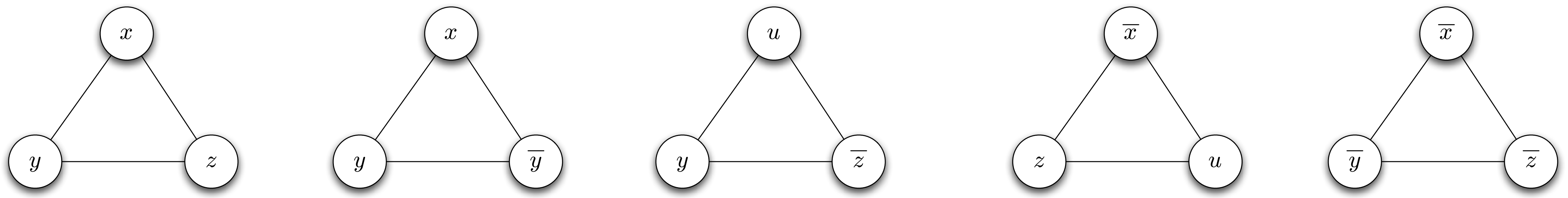
what must we do to?

3SAT $\leq_p$ INDSET

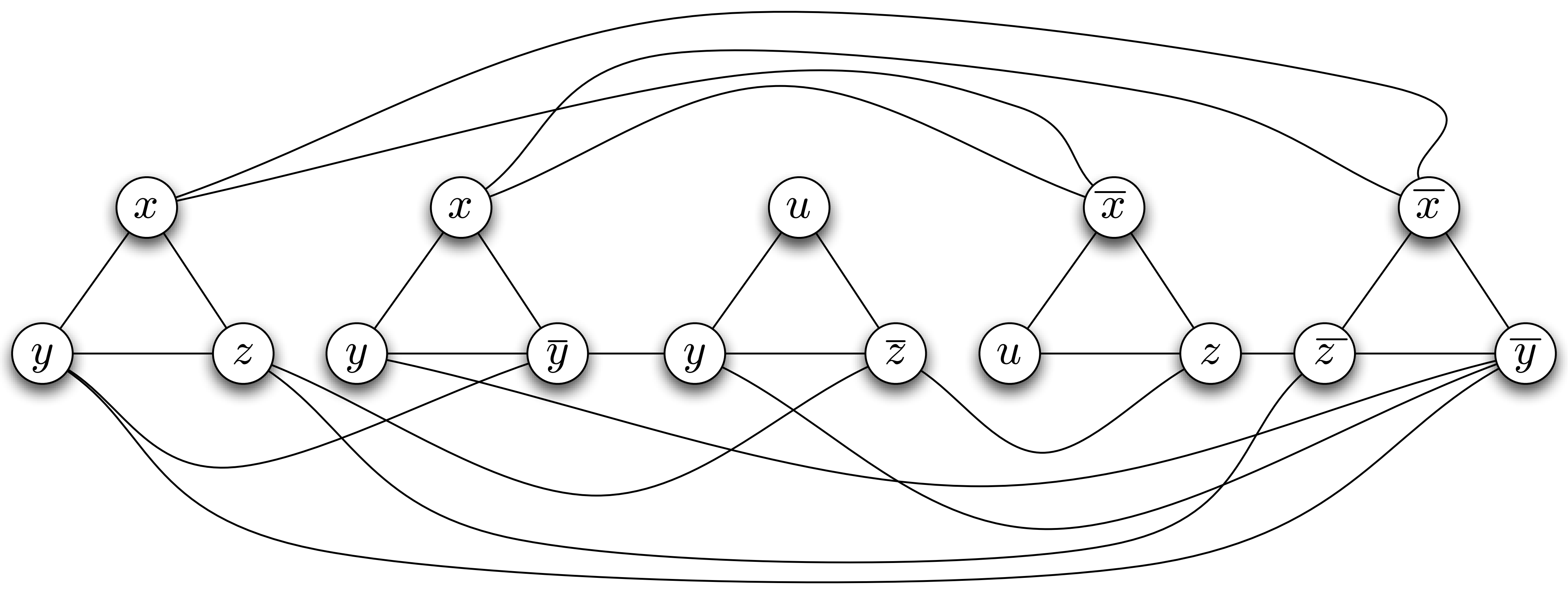
$$(x \vee y \vee z) \wedge (x \vee \overline{y} \vee y) \wedge (u \vee y \vee \overline{z}) \wedge (z \vee \overline{x} \vee u) \wedge (\overline{x} \vee \overline{y} \vee \overline{z})$$

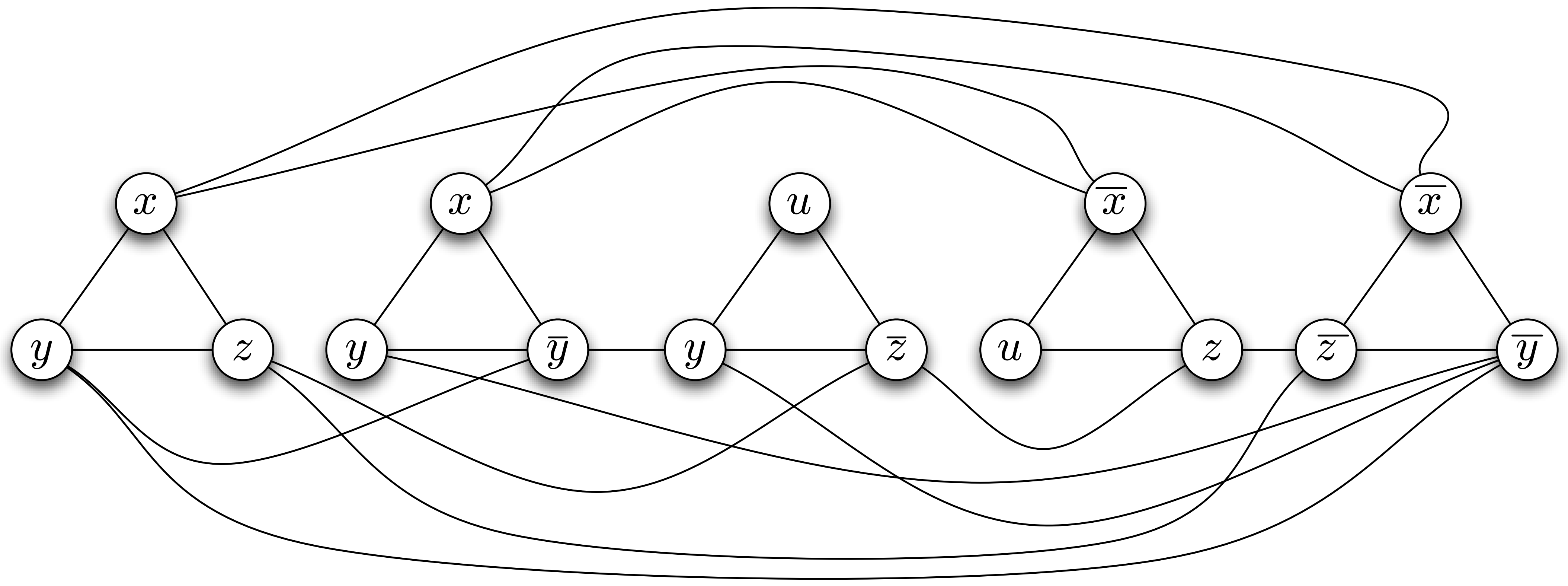
$3\text{SAT} \leq_p \text{INDSET}$

$$(x \vee y \vee z) \wedge (x \vee \overline{y} \vee y) \wedge (u \vee y \vee \overline{z}) \wedge (z \vee \overline{x} \vee u) \wedge (\overline{x} \vee \overline{y} \vee \overline{z})$$

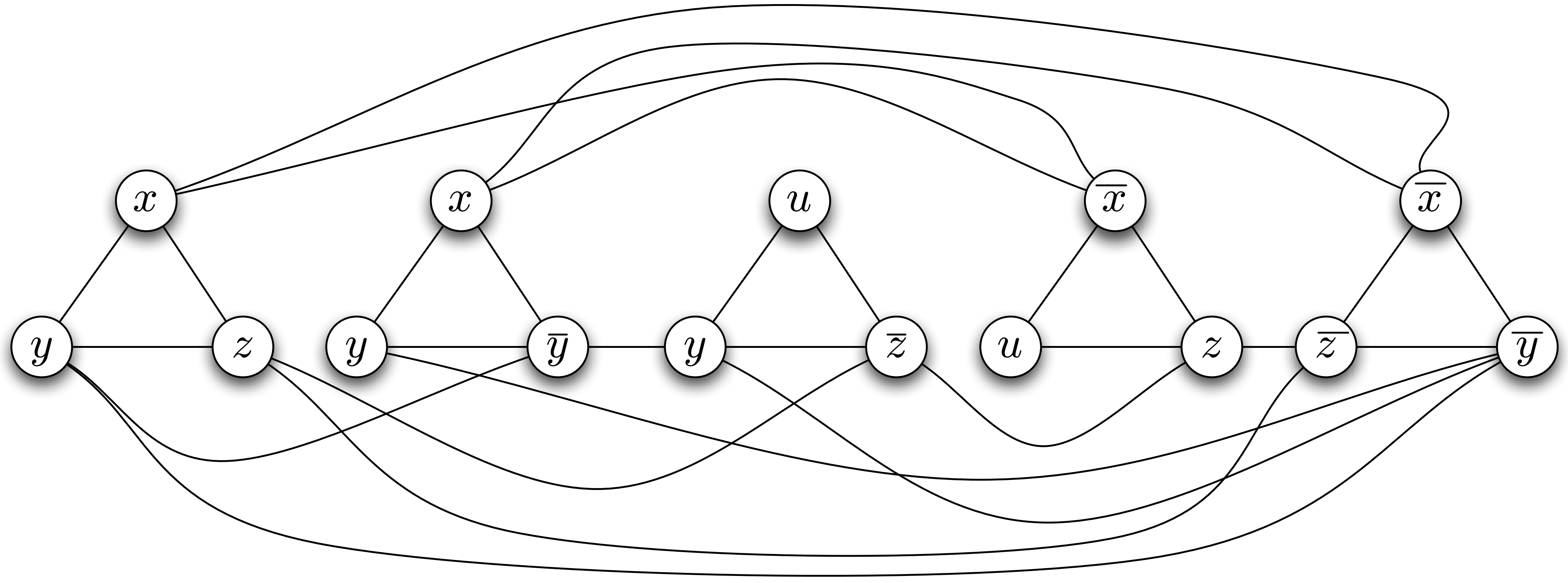


$$(x \vee y \vee z) \wedge (x \vee \overline{y} \vee y) \wedge (u \vee y \vee \overline{z}) \wedge (z \vee \overline{x} \vee u) \wedge (\overline{x} \vee \overline{y} \vee \overline{z})$$





$\phi \in \mathrm{SAT} \implies$



$(\mathsf{G}, \mathsf{k}) \in$ INDSET $\implies$

COMPLEXITY THEORY

Theory of NP

DEFINITION OF NP

A language L

DEFINITION OF NP

a language L belongs to the class NP iff

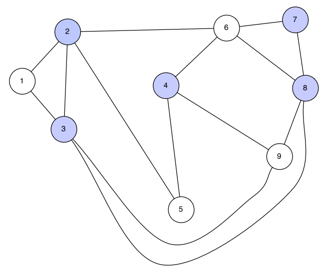
$\exists A, c$ such that

$$L = \{x \in \{0,1\}^* \mid \exists y \in \{0,1\}^{|x|^c} \text{ s.t. } A(x,y) = 1\}$$

WHY IS TRIPLETS IN NP?

$$(x_1, x_2, \ldots, x_n)$$

WHY IS INDSET IN NP?



COMPLEXITY CLASSES

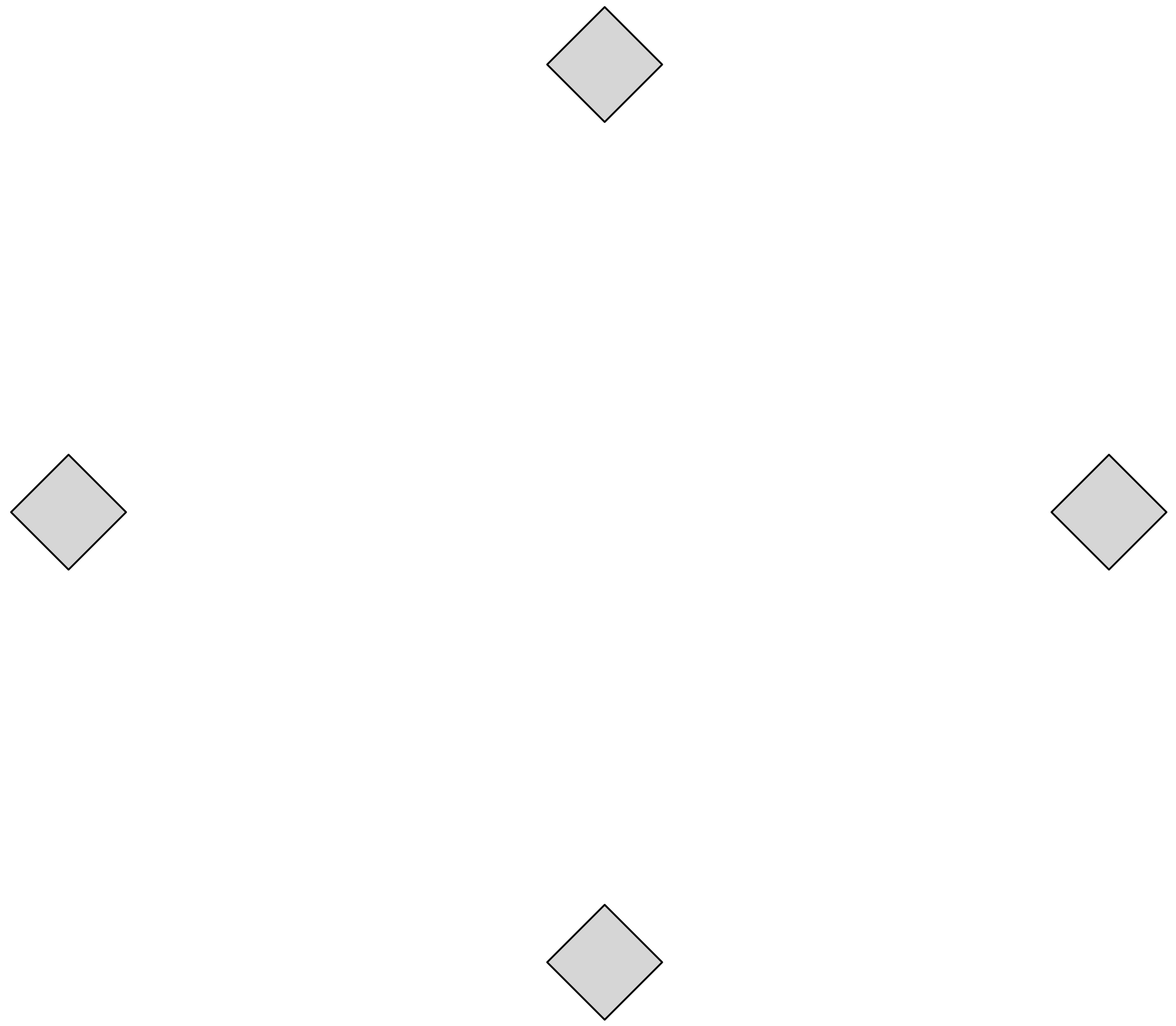
NP

P

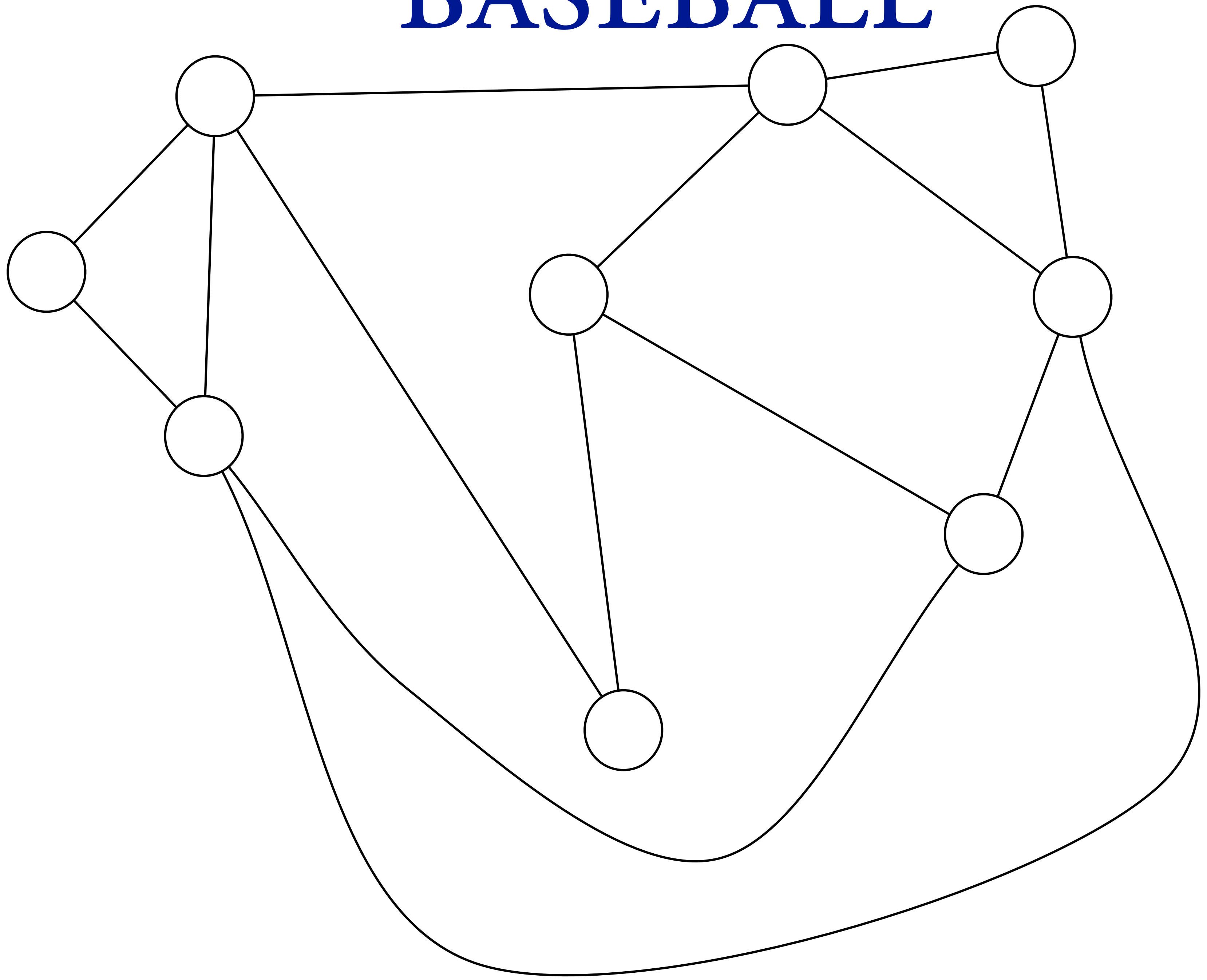
COOK-LEVIN THEOREM

The implication of this

BASEBALL



BASEBALL



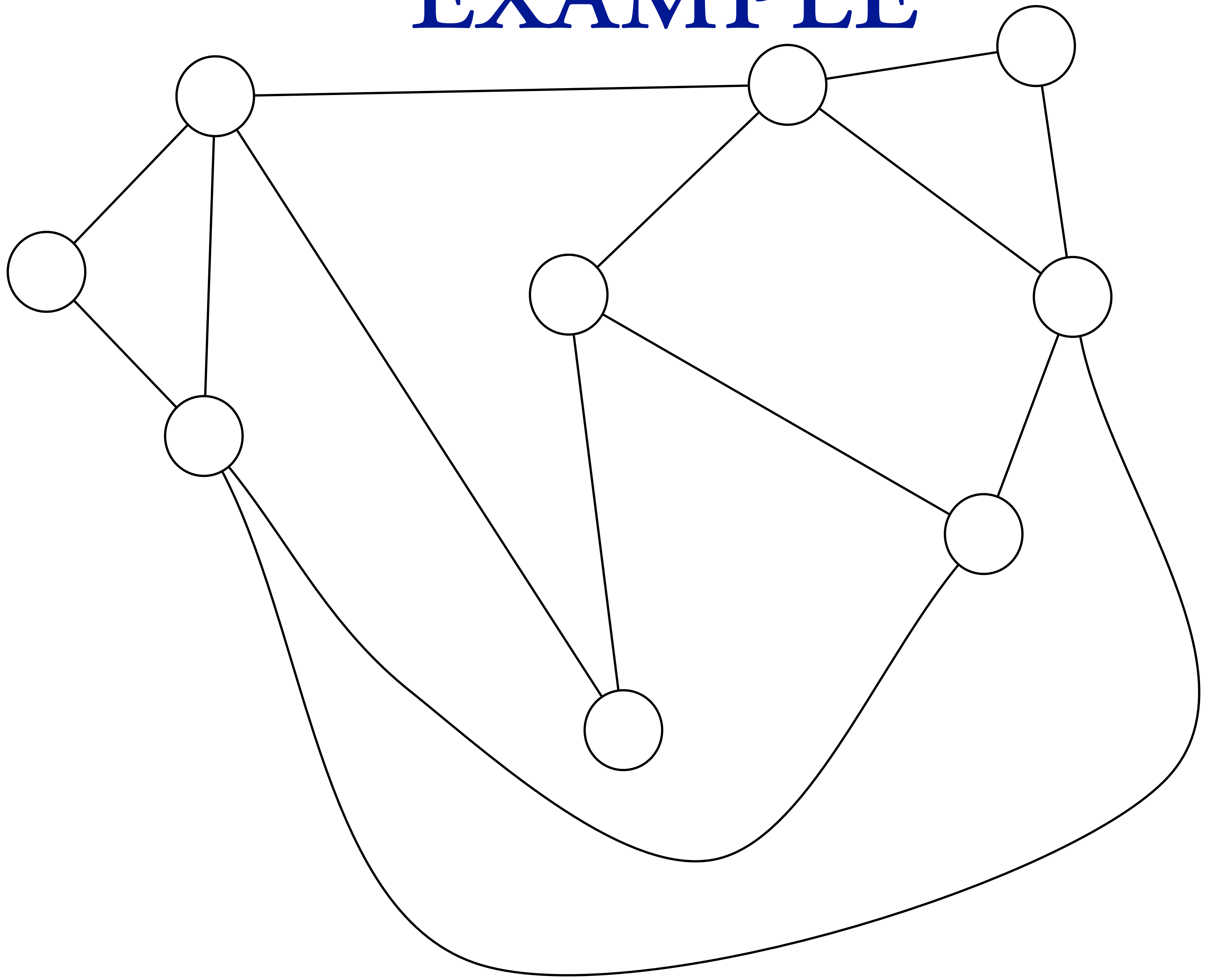
a vertex cover of a graph is a

a vertex cover of a graph is a set $C \subseteq V$

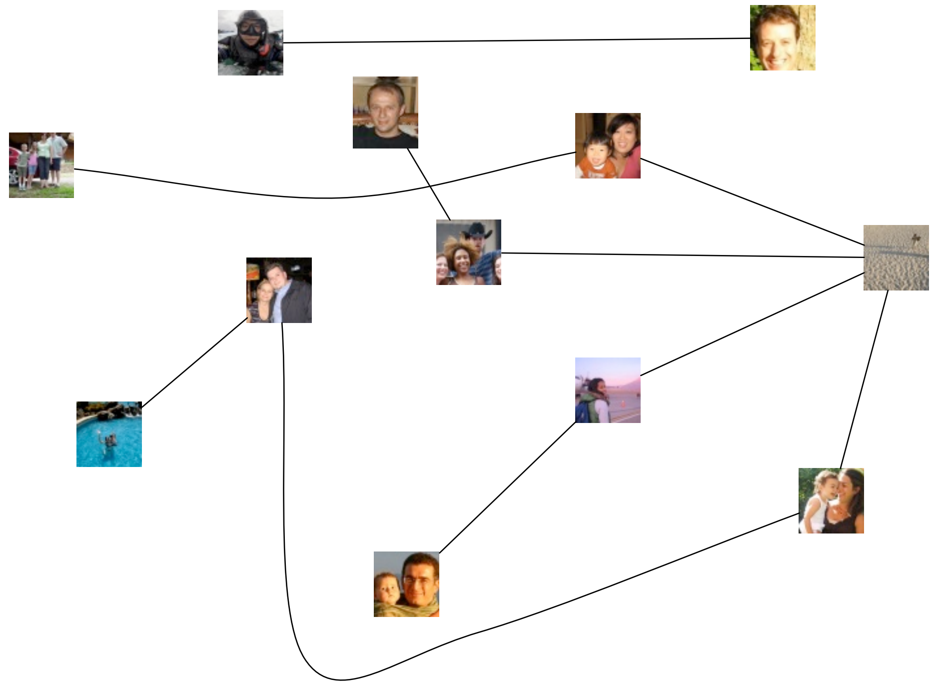
such that $\forall\ (x, y) \in E$

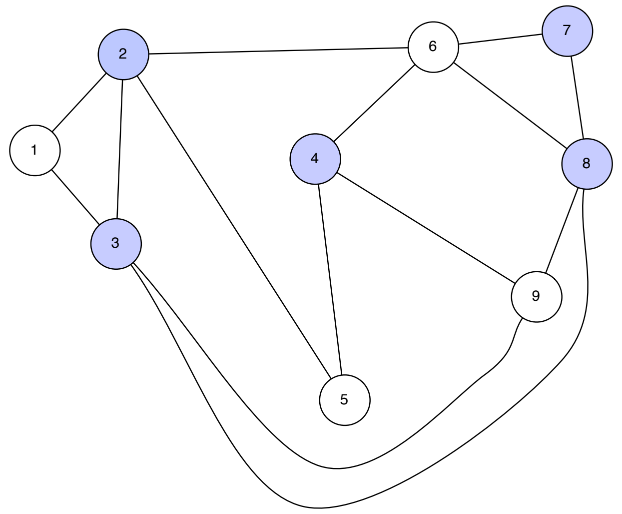
either $x \in C$ or $y \in C$

EXAMPLE



GOAL: given a graph G,





$$\text{MAXINDSET} \leq_{O(V)} \text{MINVERTEXCOVER}$$