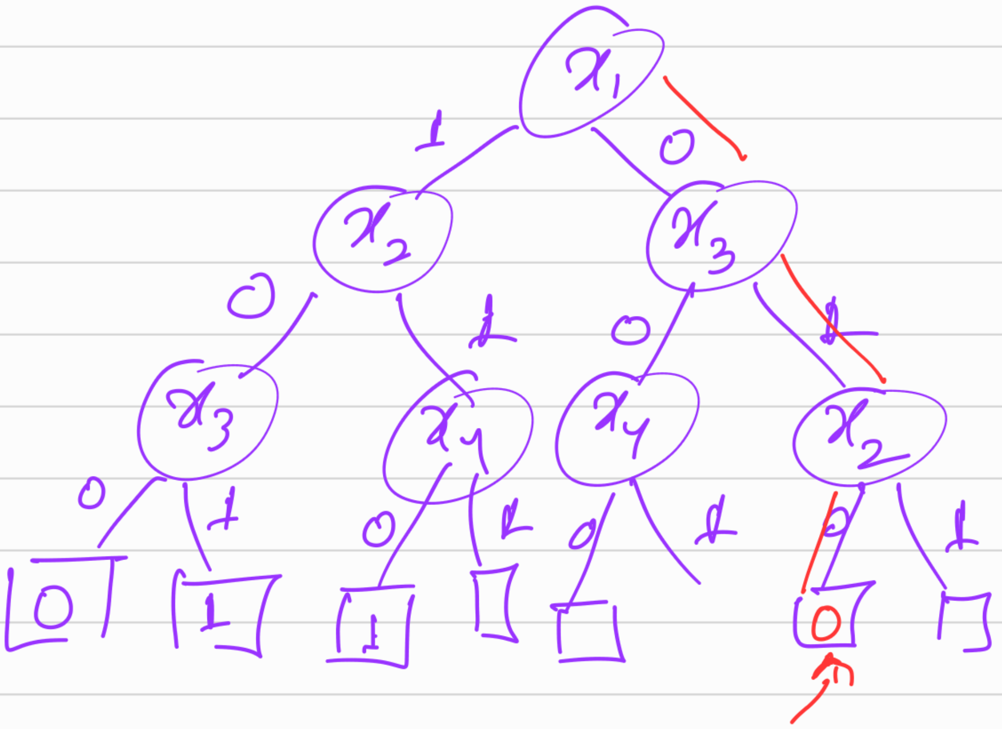
03/08/2023 Lec 2.

$f: \{0,1\}^n \to \{0,1\}$ Boolean function

→ Decision Tree.



T

$x = 0011$

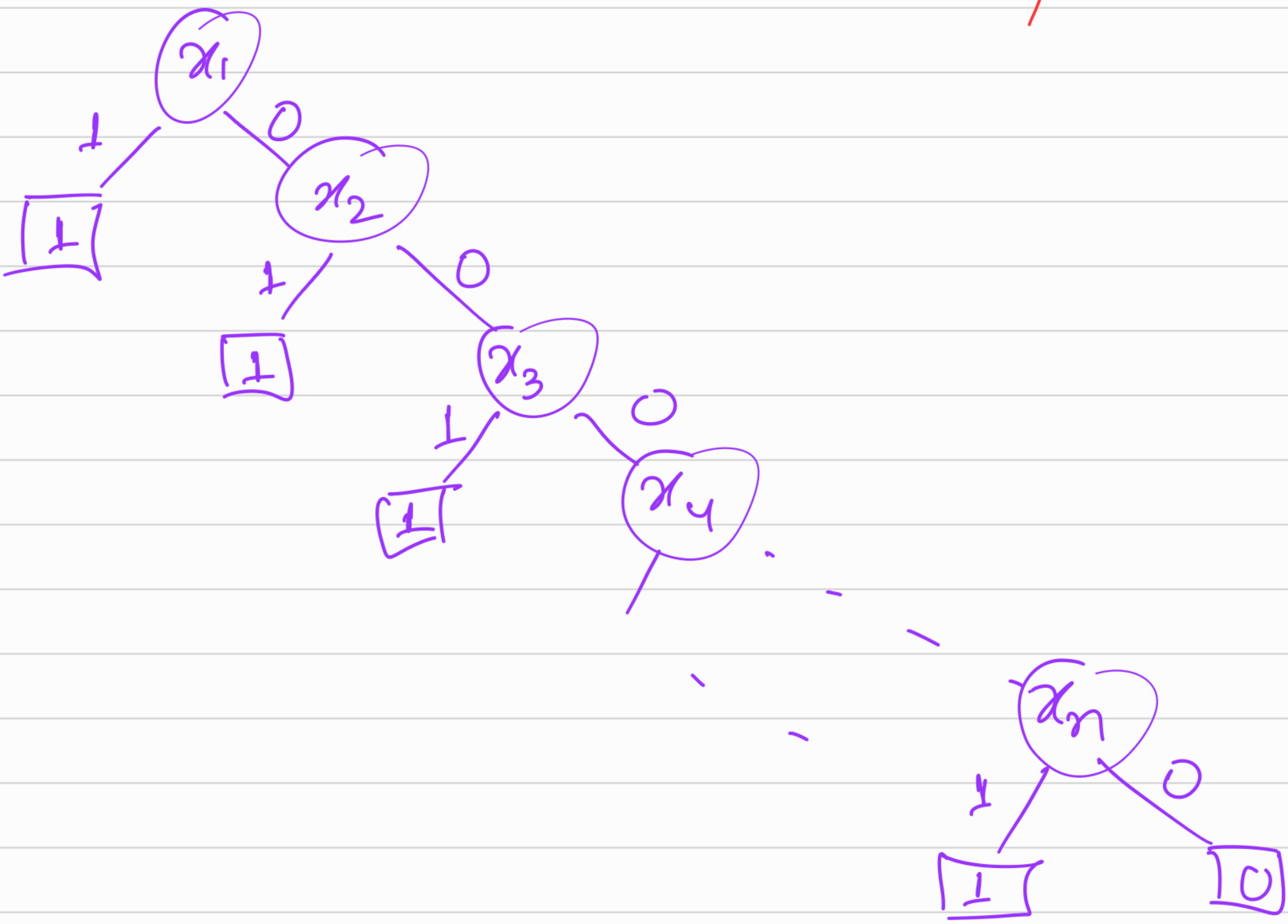
$T(x) = 0$

A Decision tree T computes a function $f : \{0,1\}^n \longrightarrow \{0,1\}$

iff $\forall\, x \in \{0,1\}^n$

$T(x) = f(x)$

$$\mathrm{OR}(x_1, \ldots, x_n) := \begin{cases} 1 & \text{if } |x| \geq 1 \\ 0 & \text{o/w.} \end{cases}$$



Prop:- if $f : \{0,1\}^n \to \{0,1\}$.
then the depth of a decision tree computing $f \leq n$.

depth of a tree = length of the longest path from the root to some leaf.

— Query complexity.

Cost of a tree T on an input x. = length of "this" unique path.

Any input x follows a unique path in the tree.

$T(x) \leftarrow$ notation.

$\text{cost}(T, x) \leftarrow :=$ cost of T on the input x.

cost of a Tree T

$$\text{cost}(T) = \max_{x \in \{0,1\}^n} \text{cost}(T, x)$$

Consider a Boolean function

$$f : \{0,1\}^n \to \{0,1\}$$

Let T be any decision tree computing f.

$$\text{cost}(T) := \max_x \text{cost}(T, x) \leq n$$

Deterministic Decision Tree Complexity

for a function $f : \{0,1\}^n \to \{0,1\}$

is $\displaystyle\min_{T \text{ computing } f} \text{Cost}(T)$

$$D(f) := \min_{T \text{ computing } f} \text{Cost}(T)$$

Prop :- Let $f : \{0,1\}^n \to \{0,1\}$

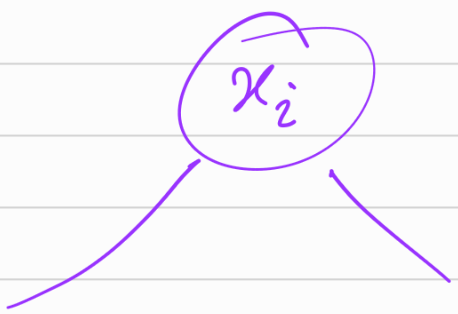
Then,

$$0 \leq D(f) \leq n$$

Ex :- $D(OR_n) \leq n$

Claim :- $D(OR_n) \geq n$

Adversary argument



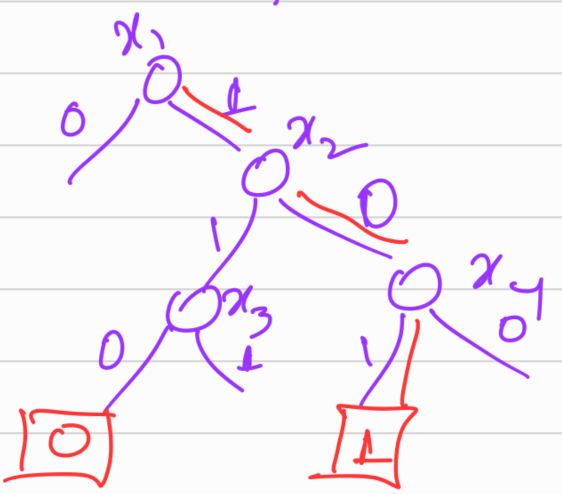
Suppose $\exists$ a tree T' of depth $\leq n-1$ that computes the Or function on n bits.

$$x_{i_1} = x_{i_2} = \cdots = x_{i_{n-1}} = 0$$

T' at this moment outputs an answer $\begin{cases} 1 & \leftarrow \text{set the last bit to } 0 \\ 0 & \leftarrow \text{set the last bit to } 1. \end{cases}$

After these $n-1$ queries you have got 2 inputs x' and y' s.t. $x' \neq y'$ and x' and y' agree on those $n-1$ bits.

but $f(x') \neq f(y')$.



$\{x_1 = 0\}$

$\{x_1 = 1,\ x_2 = 1,\ x_3 = 0\}$

$\forall y \in \{0,1\}^n$

s.t. $y_1 = 1,\ y_2 = 1,\ y_3 = 0$

$f(y) = 0$

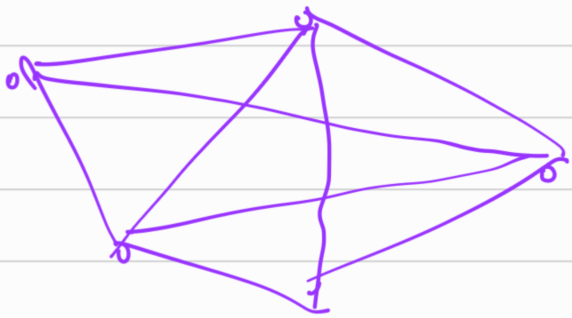
if $x \neq y$ s.t. $f(x) \neq f(y)$

then x and y must go to two different leaves.

Claim :- $D(OR_n) = n$.

$$f_{conn} : \{0,1\}^{\binom{n}{2}} \to \{0,1\}$$

$$x = \{x_1, \ldots, x_{\binom{n}{2}}\}$$

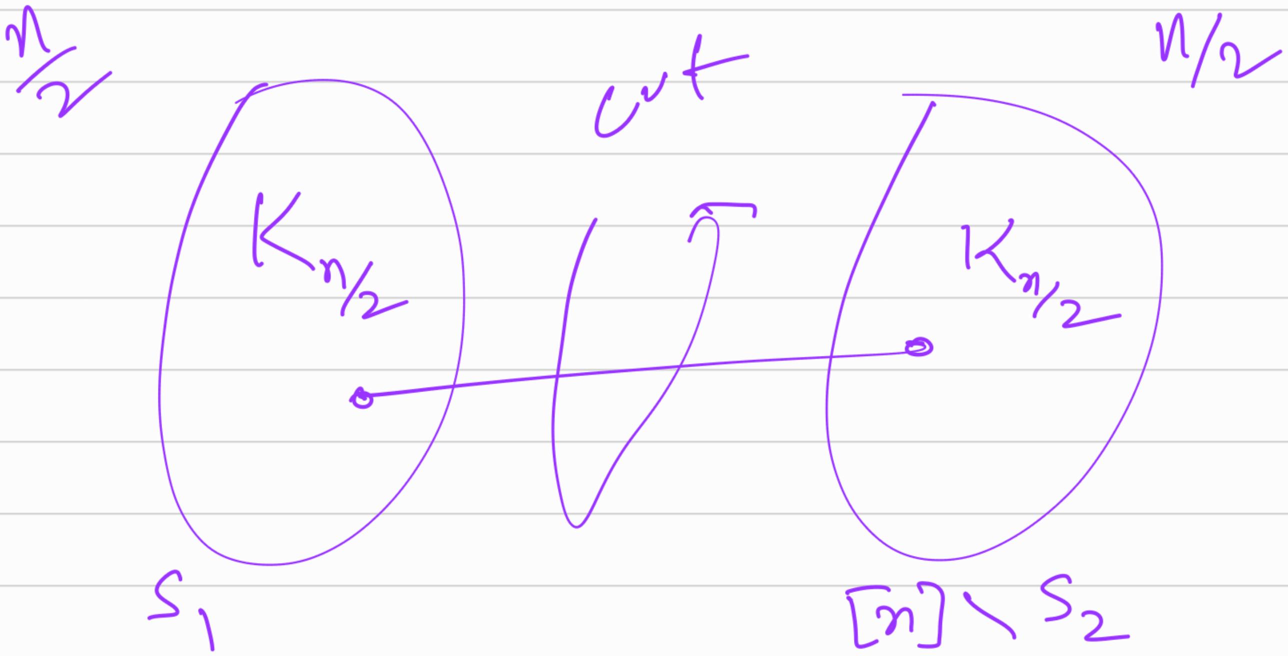


An input x corresponds to an undirected graph. G_x

$$f_{conn}(x) = \begin{cases} 1 & \text{if } G_x \text{ is connected} \\ 0 & \text{o/w.} \end{cases}$$

Thm :- $n-1 \leq D(f_{conn}) \leq \binom{n}{2}$

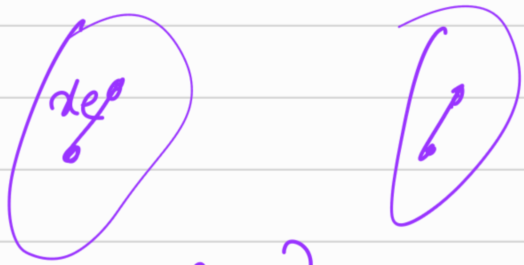
Can the lower bound be improved ?



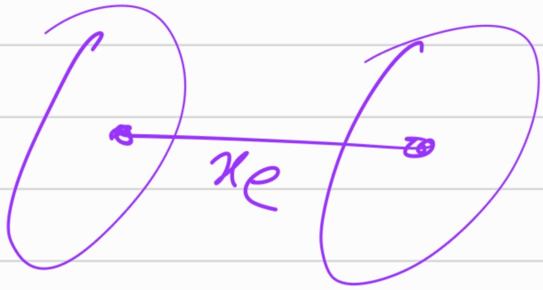
Adversaries Strategy

- Algorith queries x_e

- Adversary



$x_e \in \left\{ \begin{array}{l} \text{left hand side} \\ \text{right hand side} \end{array} \right\}$ set $x_e = 1$.

- 

set $x_e = 0$ if there are unqueried cut edges.

with this strategy adversary will force the d.t. to query every edge in the cut.

$\#$ edges in the cut $\geq \frac{n^2}{4}$

$$\Rightarrow \quad D(f_{conn}) \geq \frac{n^2}{4}$$

<u>Thm</u> :- $\quad \frac{n^2}{4} \leq D(f_{conn}) \leq \binom{n}{2}$

$$= \frac{n(n-1)}{2}$$

<u>Thm</u> :- $\quad D(f_{conn}) \geq \binom{n}{2}$

N = set of edges that are definitely not present in the graph ($x_e = 0$)

Y = present in the graph ($x_e = 1$)

set of unqueried edges. $= M$

$\triangleright$ x_{e_1} $\qquad$ $x_{e_i} \leftarrow$ query.

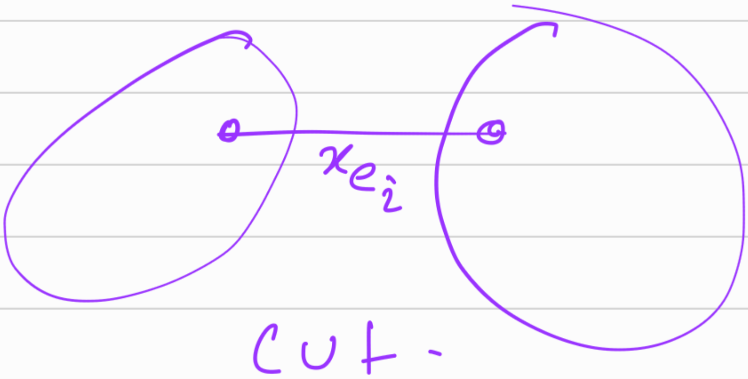
Y $\qquad$ $\left\{ \begin{array}{l} Y \cup M \\ \\ Y \cup M \setminus \{e_i\} \end{array} \right.$

if $Y \cup M \setminus \{e_i\}$ is connected

then set $x_{e_i} = 0$

o/w. set $x_{e_i} = 1$.

$Y \leftarrow$ partial graph.



cut.