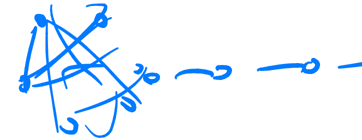
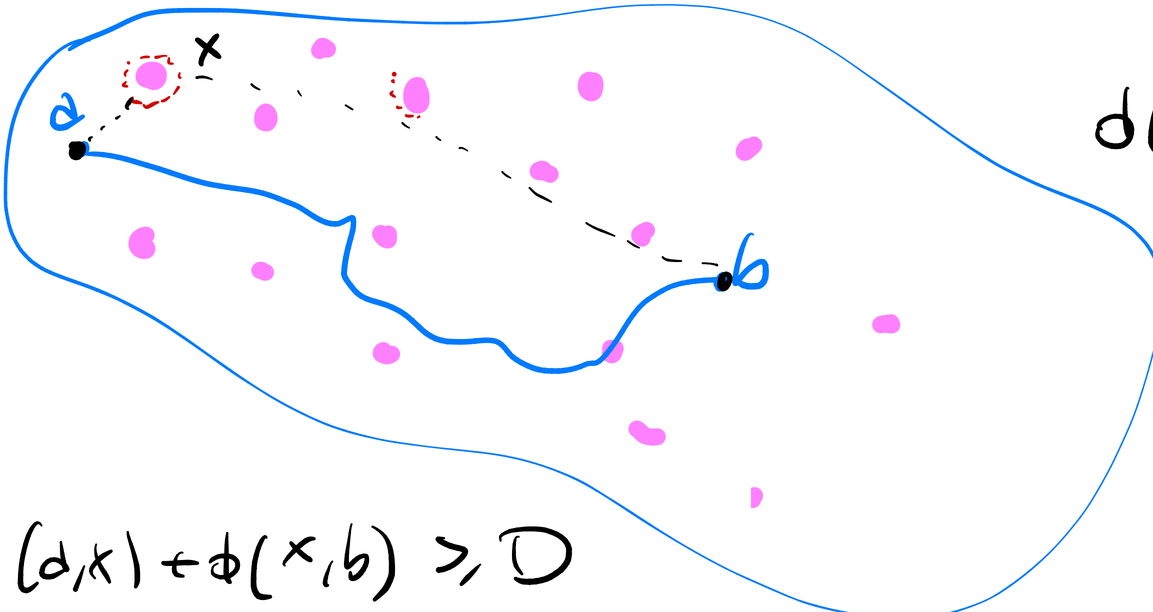
Diameter





$$d(a,b) = D$$

$$\underbrace{d(a,x)}_{\text{small}} + \underbrace{d(x,b)}_{\text{large}} \geq D$$

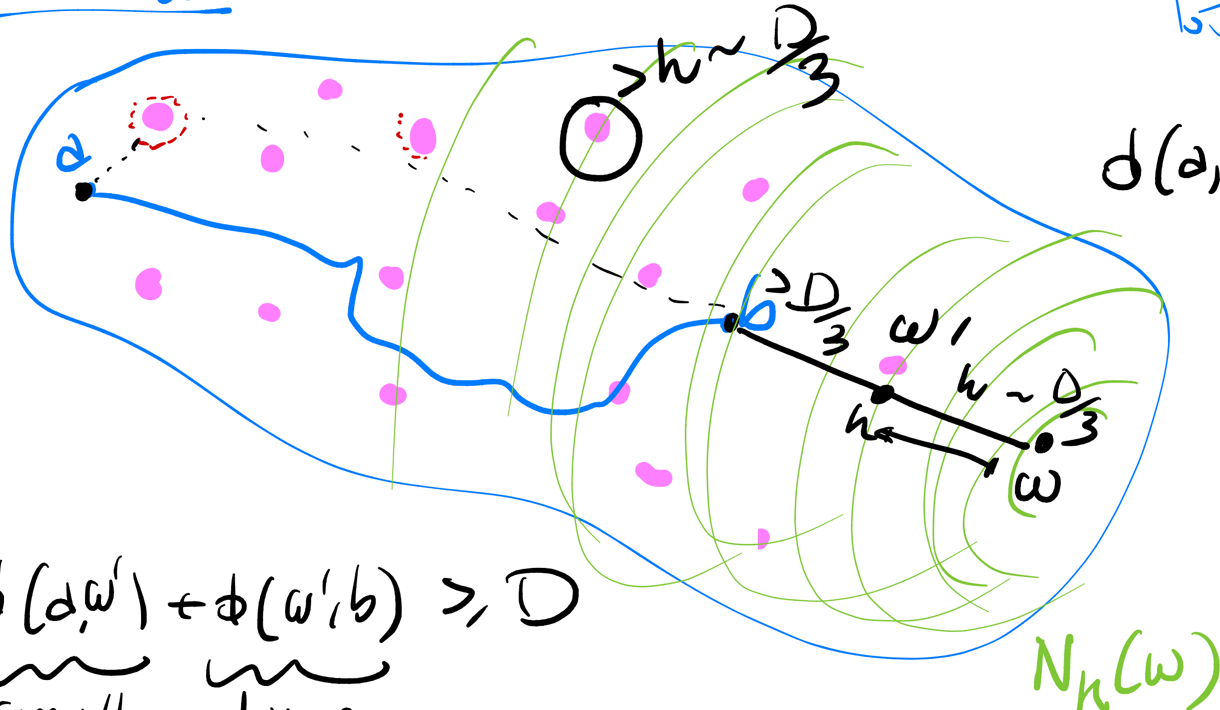
$$\Downarrow \quad d(x,b) \geq D - \overbrace{d(a,x)}^{\text{small}}$$

b.1

$K = \sqrt{n}$

Diameter

$d(a,b) = D$



b.2

$$\underbrace{d(a,w')}_{\text{small}} + \underbrace{d(w',b)}_{\text{large}} \geq D$$

$$\Downarrow$$

$$d(w',b) \geq D - \overbrace{d(a,w')}^{\text{small}} \sim$$

Can we do better than $O(n^2)$ worst case

[spoiler: NO ; and $\tilde{O}(n\sqrt{n})$ cannot be improved for approximation]

FINE-GRAINED COMPLEXITY

↳ many quadratic or cubic algorithms were not improved over decades →

CONDITIONAL LOWER BOUNDS

- difficult problem Π
- reduce Π to your problem

CONDITIONAL LOWER BOUND

- reduce SAT to the DIAMETER

SAT Boolean formulas

n' Boolean vars $x_1, x_2, \dots, x_{n'} \in \{0,1\}$

literals: x_i, $\bar{x}_i$

clauses = OR of literals $\quad x_i \vee \bar{x}_j \vee x_k$

CNF formula = AND of clauses $\quad (x_1 \vee \bar{x}_3 \vee x_4) \wedge (\bar{x}_1 \vee x_2 \vee \bar{x}_4)$

$$\begin{array}{cccccc} \cancel{0}\ 1 & 0 & 0 & \cancel{1}\ 0 & 1 & 1 \end{array}$$

assignment $\tau : [n'] \to \{0,1\}^{n'}$ $\qquad \tau: x_1 = 0,\ x_2 = 1,\ x_3 = 1,\ x_4 = 0$

τ satisfies a formula F: replacing x_i with its value $\tau[i] \in \{0,1\}$ gives $F = 1$

$\tau' = x_1 = 1,\ x_2 = 1,\ x_3 = 1,\ x_4 = 0$

Guessing τ takes $O(2^{n'} \cdot \mathrm{poly}(n'))$ time

SAT is NP-complete

$\boxed{\text{SETH}}$: Strongly Exponential-Time Hypothesis

No algorithm can solve SAT in $O(2^{\delta n'} \mathrm{poly}(n))$ time for $0 < \delta < 1$ unless $P = NP$

(check OV = orthogonal vector hypothesis)

$O(2^{n'} \mathrm{poly}(n))$ we can do

$$O\left(\big(\underbrace{2^{\delta}}_{<2}\big)^{n'} \mathrm{poly}\right)$$

$$2^{n'/3} = \left(2^{1/3}\right)^{n'}$$

We prove that $O(n^{2-\varepsilon})$ time for the DIAMETER in worst-case

$\Rightarrow O(2^{\delta n'} \mathrm{poly}(n))$ time for SAT, with $\delta < 1$ $\Rightarrow$ SETH fails

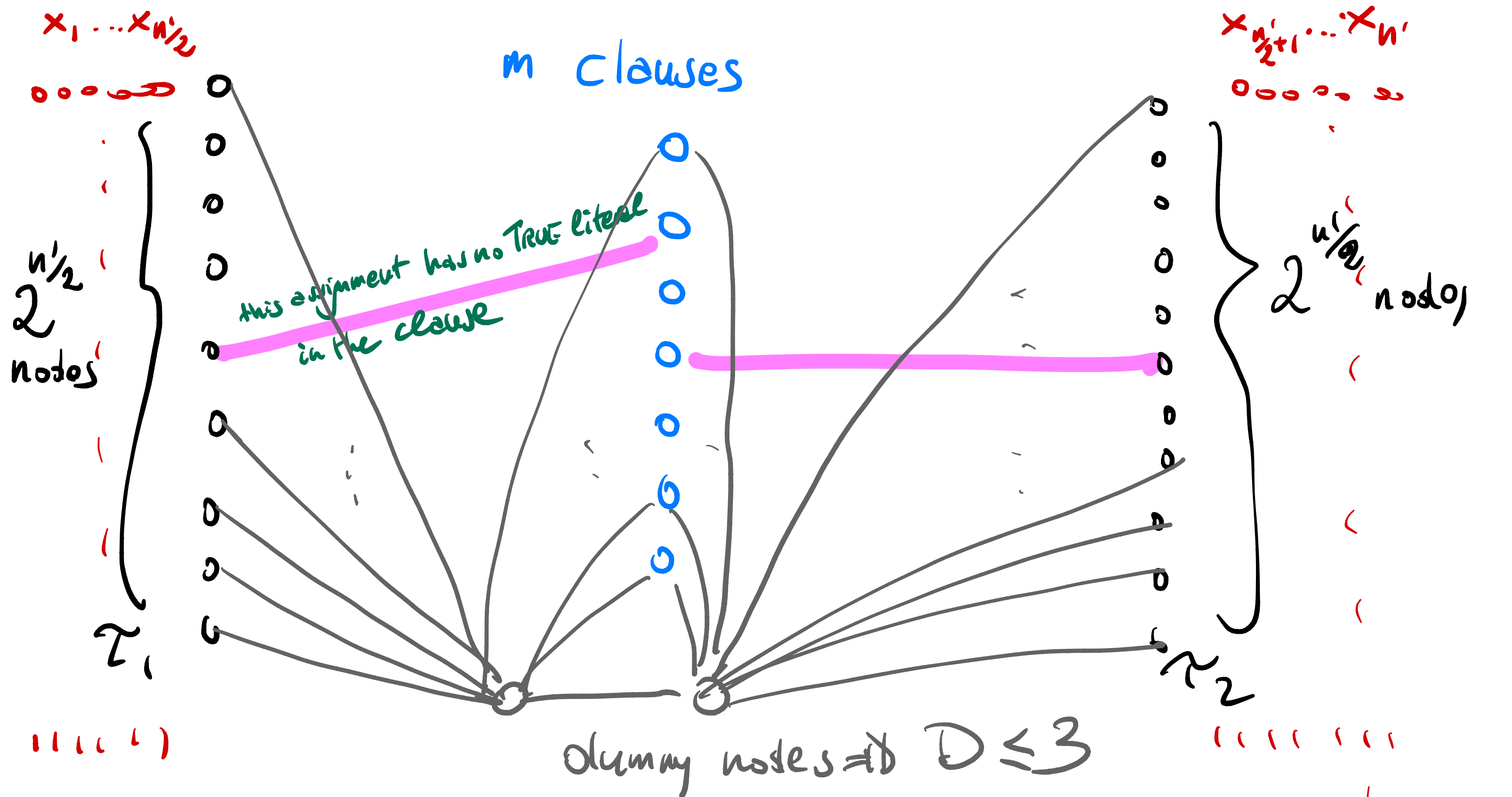
It's difficult to decide whether a graph has diameter $D=2$ vs $D=3$

$$\overset{0}{(x_1} \vee \bar{x}_3 \vee x_4) \wedge \overset{1}{(\bar{x}_1} \vee x_2 \vee \bar{x}_4)$$

τ: $\boxed{x_1 = 0,\ x_2 = 1,}\ \boxed{x_3 = 1,\ x_4 = 0}$

Given F, we build G s.t. $D=2 \Rightarrow F$ not satisfiable, $D=3 \Rightarrow F$ satisfiable

n' variables $\rightarrow$ split into 2 groups of $\frac{n'}{2}$ vars



$D=2 \Rightarrow$ for each pair of half assignments there is a pink path

<u>obs</u> $d(\tau_1, x) \leq 2$, $x \notin \tau_2$ and vice versa
<u>obs</u> $d(\tau_1, \tau_2) = 3$ without pink edges

$\Big\}$ without pink edges $D=3$

$D=2 \Rightarrow$ pink edges $d(\tau_1, \tau_2) = 2$

$\forall x \in \tau_1, y \in \tau_2 \Rightarrow d(x, y) = 2$

$\Rightarrow \exists$ clause node c s.t. $(x, c), (c, y)$ are <u>both</u> pink

$\Rightarrow$ clause c is <u>not</u> satisfied by the assignment xy

as this holds for every $x \in \tau_1, y \in \tau_2 \Rightarrow F$ is not satisfiable

<u>obs</u> F non satisfiable $\Rightarrow \forall x \in \tau_1, y \in \tau_2$: xy does not satisfy F

$\Rightarrow \exists$ clause $c \in F$ s.t. xy does not satisfy c

$\Rightarrow \exists$ pink edges $(x, c), (c, y)$

$D \leq 3$, F satisfiable $\Rightarrow D = 3$

Suppose $O(n^{2-\varepsilon})$ for D exists

$n = O(2^{n'/2})$ in our graph seen so far

$\Rightarrow D$ would be computed in $O(n^{2-\varepsilon}) = O\left(\left(2^{n'/2}\right)^{2-\varepsilon}\right)$ time

$$2^{\left[\frac{n'}{2}(2-\varepsilon)\right]} = 2^{\left[n' \cdot \underbrace{\boxed{\frac{2-\varepsilon}{2}}}_{\gamma}\right]}$$

$\gamma < 1$ as $\varepsilon > 0$

$\Rightarrow$ SETH fails

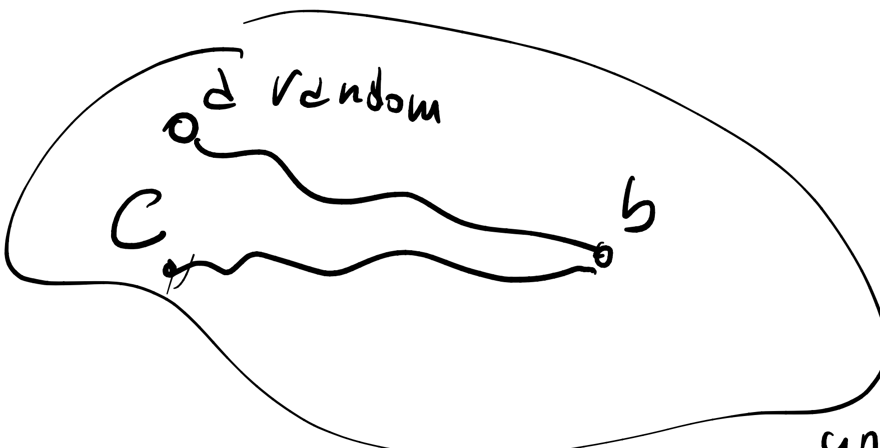
What to do in practice?

Double sweep:

- a = random node $\in V$
- b = furtest node from a
- c = furtest node from b

max$(ecc(a), ecc(b), ecc(c))$ is very close to D

repeat few times



3 BFS

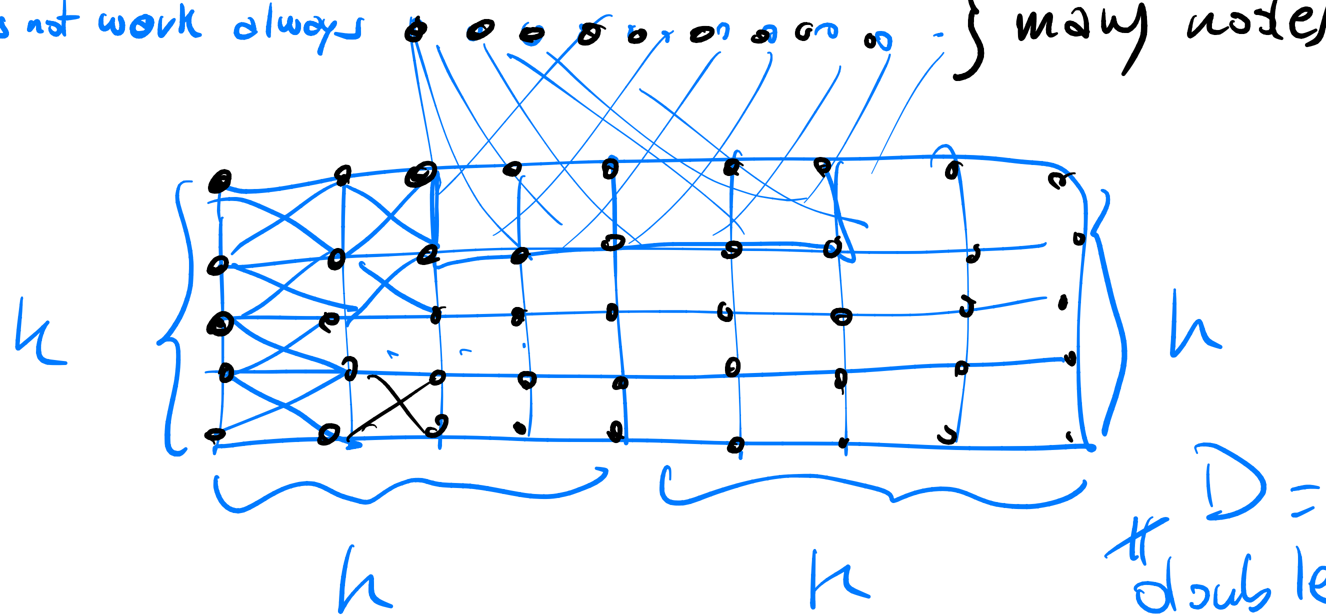
lower bound to D

upper bound $2 \times ecc(x)$

double sweep

$$\begin{cases} \text{lower\_bound} = \max(\text{lower\_bound}, ecc(a), ecc(b), ecc(c)) \\ \text{upper\_bound} = 2 \times \min\left(\frac{\text{upper\_bound}}{2}, ecc(a), ecc(b), ecc(c)\right) \end{cases}$$

does not work always



D = ?

\# double sweep = ?