- CONNECTED GRAPHS ∈ P

\[ P = \{ L \mid \text{L is decidable by a polynomial-time T.M.} \} \]

Many examples in CPSC 320 of poly-time algorithms...

\[ G \text{ graph } \rightarrow G = (V, E) \]

\[ V \text{ is a set, } E \subseteq \{ \text{pairs of vertices} \} \]

e.g. \[ V = \{ a, b, c, \alpha, \beta, \sqrt{2}, \ldots \} \]

To make sense of \( \langle G \rangle \)
we assume

\[ V = \{1, \ldots, N\}, \quad N \in \mathbb{N} \]

or -

\[ V = \{V_1, V_2, \ldots, V_N\} \]

identify with \[\{1, 2, \ldots, N\}\]

\[ \langle G \rangle \text{ with } G \text{ standard graph} \]

meaning \[ V = \{1, 2, \ldots, N\} : \]

\[ \langle G \rangle = (\# \text{ vert in base } 10) \text{ (give an edge)} \]

(give another edge)
e.g.,

\[ V = \{ 1, \ldots, 1000000 \} \]

\[ E = \{ \{ 2, 700318 \}, \{ 4, 6 \} \} \]

\[ \langle G \rangle = 1000000 \# 2 \# 700318 \]

\[ \# 4 \# 6 \]

Caution: If

\[ n = \text{length } \langle G \rangle \]

\[ |V| \text{ could be } 10^n \]

e.g. \[ V = \{ 1, \ldots, 1000 \}, E = \emptyset \]

\[ \langle G \rangle = 1000 \]
We could write

$$
\langle G \rangle = \text{(\# of vertices)} \begin{pmatrix}
1 & 0 & 1 & 0 & 0 & 0 & 1 \\
0 & 1 & 0 & 0 & 0 & 0 & 1 \\
1 & 0 & 1 & 0 & 0 & 0 & 1 \\
0 & 1 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1
\end{pmatrix}
$$

N
N x N adjacency matrix

$$
\langle G \rangle = \begin{pmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix}
$$

adjacency matrix

Optim:

# vertices is 17, could write

17 is base 10

Unary --- to follow in 2 weeks
**CONNECTED-GRAPHS**

\[ \{ \langle G \rangle \mid G \text{ is a (standard) graph s.t. for any 2 vertices of } G, \text{ there is a path between them} \} \]

Connected

Not connected

Path from \( V_1 \) to \( V_2 \)
How we claim:

CONNECTED - GRAPHS \in P

Algorithm: pick any vertex
\forall V = \{1, \ldots, N\}:

\begin{itemize}
  \item which vertices are connected to \( V \)
  \item extra vertex
\end{itemize}
Say that \( N = 10^6 \)

\( E = \emptyset \)

\( V = 1 \)

\( v \) is \( V = 2 \) connected to \( v = 3 \) \( \cdots \)

\( \vdots \)

\( \vdots \)

\( = 10^6 \) \( \cdots \)

Algorithm, to be poly-time, has to search through the edges not \( \cdots \) vertices
$NP = \text{non-deterministic poly-time}$:

Turing machine: $(Q, \Sigma, \Gamma, \delta, q_0, q_{acc}, q_{rej})$

$sit. \quad \delta : Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R\}$

$k$-tape TM: same, but

$\delta : Q \times \Gamma^k \rightarrow Q \times \Gamma^k \times \{L, R, S\}^k$

non-deterministic TM: (1-tape)

$\delta : Q \times \Gamma \rightarrow \text{Power}(Q \times \Gamma \times \{L, R\})$

$a \rightarrow b, L$

$\delta(\varnothing) \rightarrow \delta'(\varnothing')$
could be nothing:

\[ a \rightarrow b, r \]
\[ a \leftarrow U, l \]
\[ a \rightarrow a, n \]

nothing to do if you read it.
OK

initial config:  \( q_0 \)  \[
\text{a b l b a l n u u u ...}
\]

---

Accepting somewhere

then the non-det computation accepts the input
Time : A non-deterministic TM, \( M_j \), takes time \( \leq f(n) \), where \( f : \mathbb{N} \rightarrow \mathbb{N} \), if all possible computation paths, on any input of length \( n \), halt or stop in \( \leq f(n) \) steps.
$3$-COLOR = $\{ \langle G \rangle \mid G$ is a graph that is $3$-colourable $\}$

$3$-colourable: colour the vertices with $3$-colours

Formally, $V \rightarrow \{ \text{red, green, blue} \}$

s.t. each edge has different colours at its
Endpoints.

E.g., red

blue

green

red blue

3-colorable

red

red
3-COLOR can be decided by a poly-time non-deterministic algorithm:

Start 1

2

3

6

4

5

6 vertices

Write down a sequence

Red Red Blue Green Green Blue

1 2 3 4 5 6