Directed graphs $G = (V, E)$ with ...

exponentially many vertices ???

... $n$ = reasonable number

algorithms running in poly($n$) time

poly($n$) space

Might happen in PSPACE, PTIME,

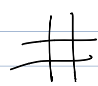
NPSPACE, NPTIME

=

e.g. Turing machine using $2n^5$ cells (or less)

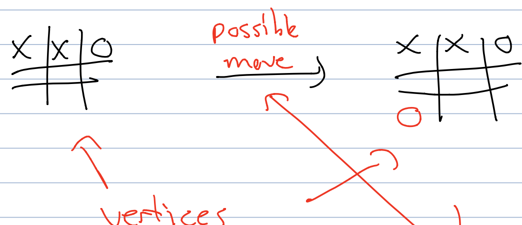
\# Configs is $C^{(2n^5)}$

=

Tic-tac-toe : 9 cells  , # configs $\leq 3^9$

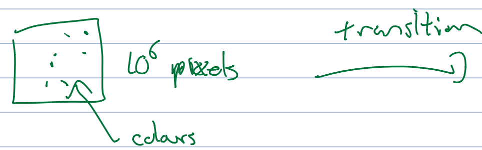
4×4×4 " : 64 cells , ... $\leq 3^{64}$

So sort of directed graph




tic-tac-toe digraph

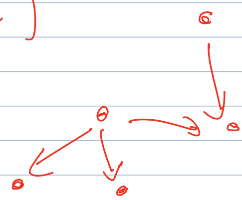
Video game




=

BTW: On the homework : $\cancel{B \geq n+m}$ → $B \geq 2n+m$

$G = (V, E)$



vertices $\subset \Lambda^N$

alphabet ($\Lambda$) — think of it as polynomial ($N$)

Assume:

(1) Given $s \in \Lambda^N$ string, we check is $s \in V$

in time polynomial $n$, $n$ some parameter;

$N \le \text{poly}(n)$

(2) Given $s_1, s_2 \in V \subset \Lambda^N$, we can tell if

there is an edge $s_1 \longrightarrow s_2$ (in poly(n))

---

Given $S_{init}, S_{term} \in V \subset \Lambda^N$,

write

$$\text{ThereIsPath}(S_{init}, S_{term}, T) = \begin{cases} \text{true} & \text{if there is a path in } G \text{ from } S_{init} \text{ to } S_{term} \text{ of length } T \text{ or less} \end{cases}$$

$$S_{init} \underbrace{\nearrow \searrow \nearrow \searrow \nearrow \longrightarrow}_{\le T \text{ steps}} S_{term}$$

=

Recursive alg to see if $\text{ThereIsPath}(S_{init}, S_{term}, T)$:

~~Pseudocode~~:

(1) for all $s \in V \subset \Lambda^N$ ← go thru $s$ in lexicographical order

$s = 13719218$
$13719219$

Recursively test:

$1^{st}$ → $\text{ThereIsPath}(s_{init}, s, \lfloor T/2 \rfloor)$ and ← true or false

if true reuse space and do → $\quad '' \quad '' \quad '' \quad (s, s_{term}, \lceil T/2 \rceil)$

(think of $T$ as $c^n$, maybe $|\Lambda^N|$)

SPACE need for $\text{ThereIsPath}(s_{init}, s_{term}, T)$: say $f(T)$

$s_{init} \longrightarrow s \longrightarrow s_{term}$

(1) ← remember $s$ and step thru it, space $N$, $|\Lambda|$ symbols

$T \geq 2$:

$f(T)$ = space $N$, $|\Lambda|$ symbols + const

$$f(T) = N + \text{const} + f(\lceil T/2 \rceil)$$

$$f(T) \leq f(\lceil T/2 \rceil) + cN$$

$$\leq f(T/4) + cN + cN$$

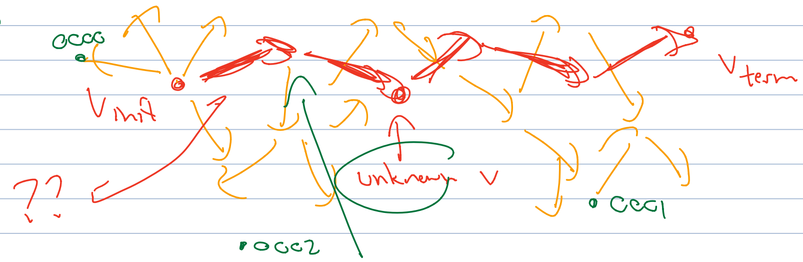
$$\leq \dots \leq cN \log_2(T) + f(1)$$

parameter

$f(1)$: reasonable, poly in $n$

(time would be $f(T) \leq cN \cdot T$)

Algo




$T = 5$

$\lfloor T/2 \rfloor = 2$

$\lceil T/2 \rceil = 3$

$T = 1$ $\quad \circ \xrightarrow{??} \circ$ } part of the rules/assumptions ...

=

Next: Want a PSPACE complete problem (§ 8.3)

Going to be sneaky.

NP-complete complete language:

NP-SNEAKY $= \{ \langle M, w, 1^t \rangle \mid$ The non-deterministic TM $M$ accepts $w$ within time $t$ (there is an accepting path) $\}$

$= \langle M, w, 1^{30} \rangle$

12#...1 $\quad$ 3#5#1#2 $\quad \underbrace{1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1 \ldots 1}_{30 \text{ times}}$

unary = "base 1"

extremely

SAT NP-complete

① SAT ∈ NP easy

(2) $L \in NP \Rightarrow L \leq_p NP$ difficult

NP SNEAKY

① more difficult

② very easy