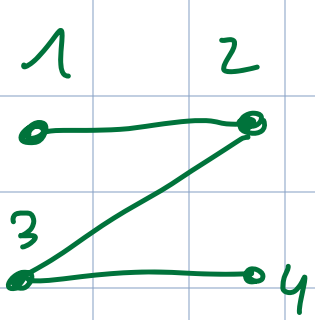
Lecture 20

Hw 5: tomorrow, due the 30th.

↳ start early, it looks short but isn't

tiger m edge

To encode a graph G (n vertices):



(1) Adjacency list:

$\{(1,2), (3,4), (2,3)\}$

size is $O(m \cdot 2 \log n) \leq O(n^2 \log n)$

(2) Adjacency matrix

	1	2	3	4
1	0	1	0	0
2	1	0	1	0
3	0	1	0	1
4	0	0	1	0

$M_{ij} = 1$ iff (ij) is an edge

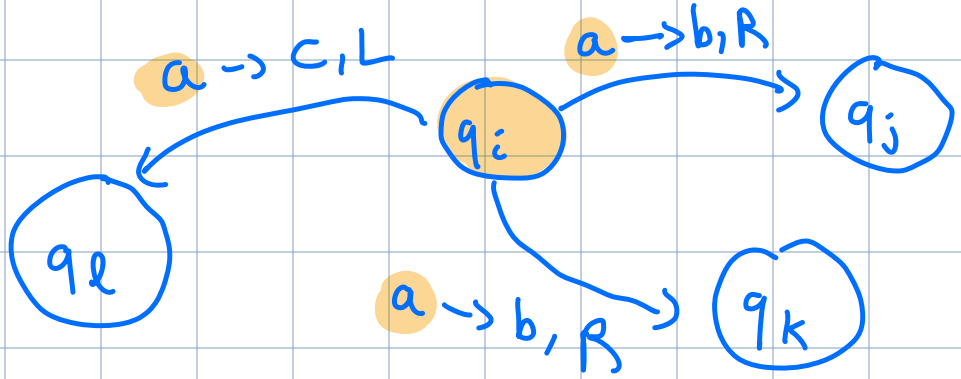
$O(n^2)$

Today : NP 7.3 + 7.4 ?

Some refresher

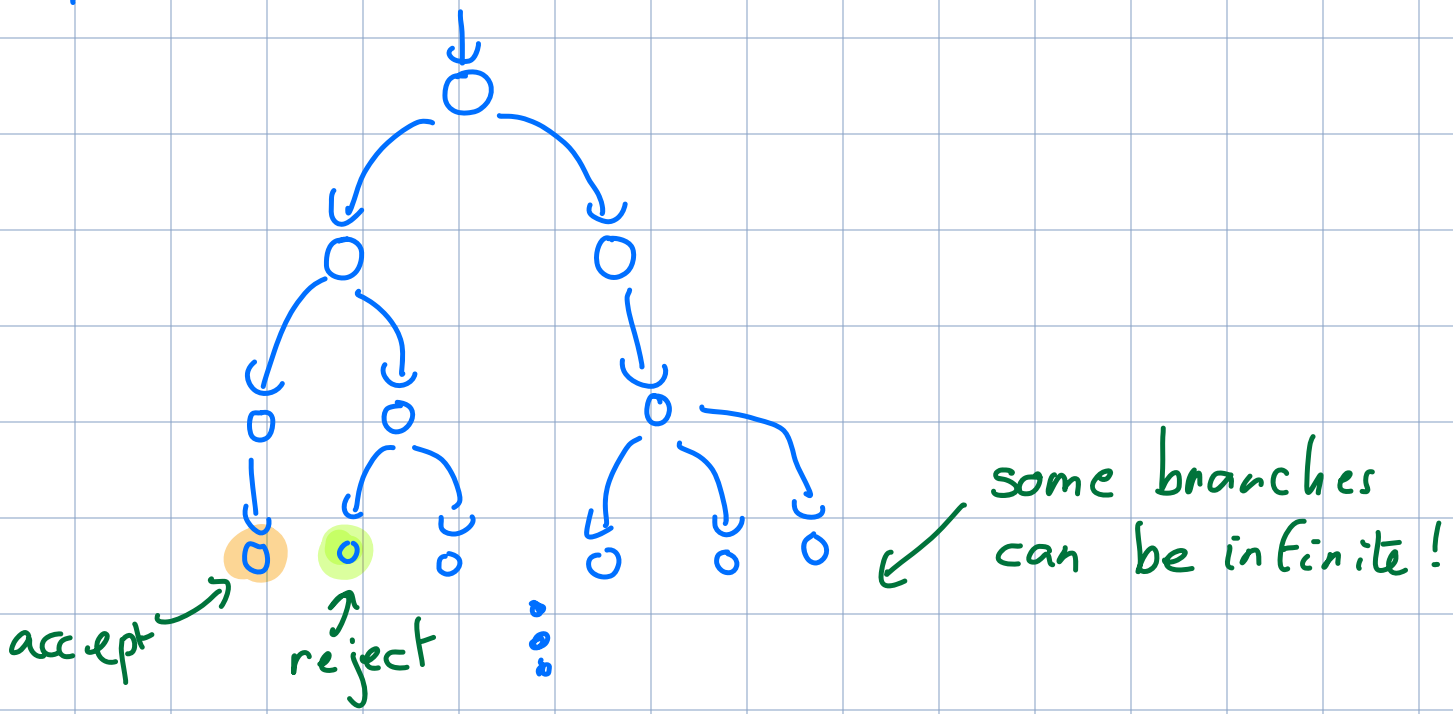
An NTM is a nondeterministic TM:

- at each step 0, 1 or more transitions possible



at q_i & read a "a"
3 transitions !

Just like an NFA: on input w many branches!

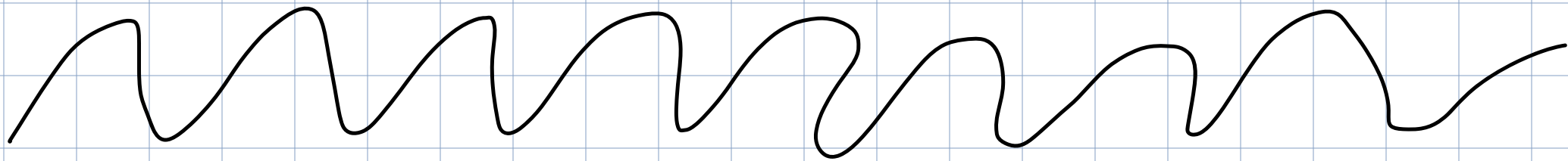


An NTM accepts w if some (at least 1) branch ends in accept

Else: it rejects w (branches reject / go on forever)

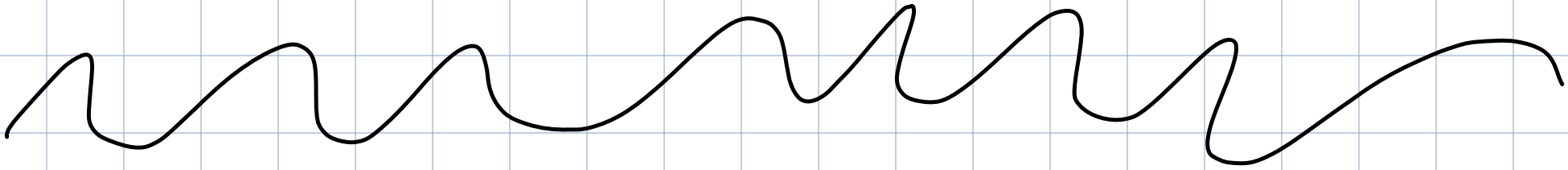
An NTM N decides a language L if:

- $L(N) = L$
- on every input $w \Rightarrow$ all branches halt
  (the computation tree is finite)



Let N be an NTM decider. The running time of N is $f: \mathbb{N} \to \mathbb{N}$ where

$f(n) = \max_{w,\ |w|=n}$ max # of steps any branch of N takes on w before halting

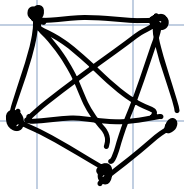


NP: When can you guess a solution & check it's correct.

For lots of pbs $\Rightarrow$ no idea if they are in P

$\hookrightarrow$ we need to search over $\approx 2^n$ # of solutions

Ex: $Clique = \{ \langle G, k \rangle \mid G$ is a graph with a k-clique $\}$



Naive algorithm: For each subset S of k vertices check if it's a clique in G

↳ if find a k-clique accept

↳ else reject.

$\binom{n}{k} \simeq O(n^k)$

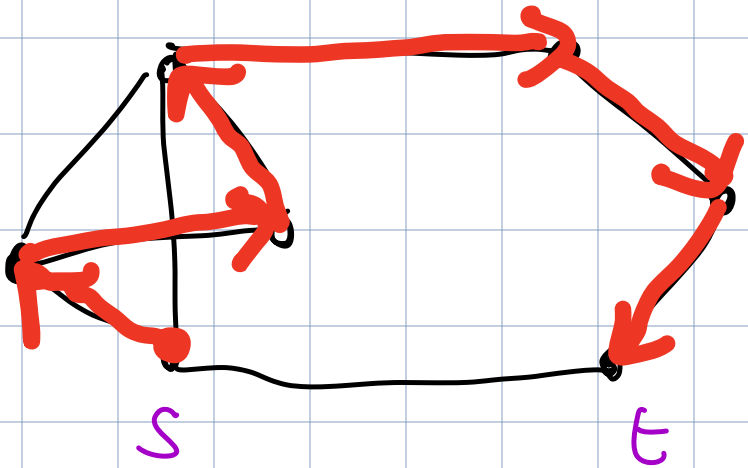
$k = n/2$

$O(n^{n/2})$ => exponential time!

But for this problem if you give

me a "proof" => a set of k vertices that forms a clique => easy to check.

Ham path = { <G, s, t> | G is an undirected graph & there's a path from s to t visiting each vertex exactly once }



Bad algo: try all possible paths from s to t => check if

one is Hamiltonian

$\approx O(n!)$

NP: it's easy to verify a solution

Def 1: NP = { L | L is a language decided by an NTM running in polynomial time }

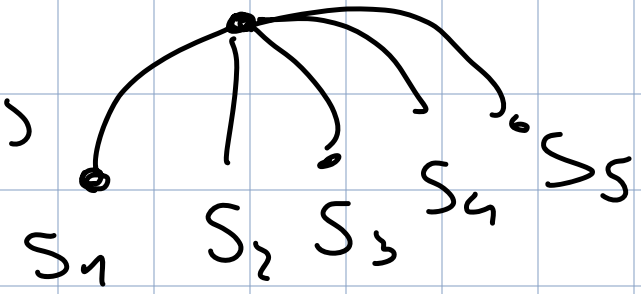
Thm: $P \subseteq NP$

Idea: NTM guesses a solution +

check if's connect

Ex Clique is in NP

On input $\langle G, k \rangle$:



(1) Nondeterministically pick k vertices from G into a set S

(2) For each $u, v \in S$: ($u \neq v$)

- (4) check that (u,v) is an edge in G
- (5) If not reject

S is a k-clique

O(1) ← (6) Accept $\langle G, k \rangle$

(1) NTM runs in poly time : input is $\langle G, k \rangle$

so input size is $O(n^2 + \log k)$

Pick k vertices from S: takes poly (n)

(if $k > n \Rightarrow$ just reject)

so $k \leq n$

lines (2) - (5) : $\leq |S|^2$ pairs to look at

$\leq n^2$

checking if (u, v) is an edge

↳ check $\langle G \rangle$ ⇒ poly(n) time.

line (6) ⇒ O(1) time

So running is poly(n) which polynomial in $|\langle G, k \rangle|$

(2) If $w \in L$, some branch/path of the NTM leads to accept.

If $\langle G, k \rangle \in$ Clique ⇒ then ∃ a clique S^* in G of size k.

So on line 1, if I pick $S \leftarrow S^*$,

lines (2) - (5) will pass (all edges are there since S^* is a clique)

So my NTM accepts.

(3) If $w \notin L \Rightarrow$ <u>every</u> branch rejects

If $\langle G, k \rangle \notin$ Clique $\Rightarrow$ there's no k-clique in G. So no matter what S I pick $\Rightarrow$ some edge will be missing

So lines (2) - (5) will reject.

An equivalent def of NP:

A verifier for a language $L \subseteq \{0,1\}^*$ is an algorithm V — a deterministic TM

$L = \{w \mid V \text{ accepts } \langle w, c \rangle \text{ for some string } c\}$

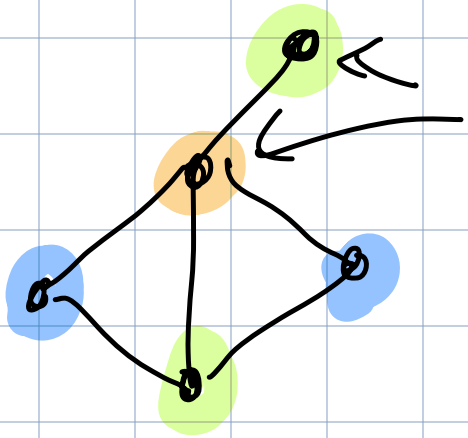
Idea: "c is a proof that $w \in L$"

A verifier is poly time if if runs in polynomial time in $|w|$ (not $|w|+|c|$)

↳ to be polytime need $|c| = \text{poly}(|w|)$

Def 2: $NP = \{L \mid L$ has a polytime verifier $\}$

Ex: k-coloring $= \{\langle G, k \rangle \mid$ for each vertex you can assign it a colon $1, \ldots, k$ such that no adjacent vertices have the same color$\}$



can't be the same color!

(1) The verifier runs in poly time:

↳ input size is $O(n^2 + \log k)$

↳ $\langle G, k \rangle$

The size of c is $O(n \cdot \log k)$

since it's a list of n numbers between 1 & k

↳ line (1) takes poly time

└, line (3) : checking $c_i = c_j$

⇒ poly (n, log (k)) time

So the loop runs at most $O(n^2)$ times (max # of edges)

So the verifier runs in polynomial time in $|\langle G, k \rangle|$

(2) If $w \notin L$, $V(w, c) \Rightarrow$ reject for all c.

If $\langle G, k \rangle \notin L$:

- If c has the wrong format
  $\Rightarrow$ reject

- Else because G is not k-colorable
  c must color 2 adjacent vertices
  the same color $\Rightarrow$ Algo rejects.

So Algo always reject $\langle \langle G, k \rangle, c \rangle$
when $\langle G, k \rangle \notin L$.

(3) If $w \in L$, $V(w, c) \Rightarrow$ accepts

for some c.

If $\langle G, k \rangle \in L$, there is a k coloring c^* of G. So when c is such that c_i = color of vertex i in c^*

I claim $V(\langle G, k \rangle, c)$ will accept.

Indeed $\forall$ (ij) that are edges of G, i & j have a different color in c (because of c^*)

So lines (2) - (4) do not reject

& the verifier accepts.

Equivalence between NTMs & verifier definition of NP.

$\mathcal{L}_1 = \{L \mid L$ is accepted by a polytime NTM$\}$

$\mathcal{L}_2 = \{L \mid L$ has a polytime verifier$\}$

WTS $\mathcal{L}_1 = \mathcal{L}_2$.

a) $\mathcal{L}_1 \subseteq \mathcal{L}_2$. Let $L \in L_1$ w.T.s $L \in L_2$

Let N be an NTM for L_1 running in time n^k.

↳ for L_1

Verifier: on input $\langle w, c \rangle$:

$O(n^k)$ (
- Simulate N on input w. treat each symbol of c as the next nondeterministic step to take
- If this branch of ∨ accepts ⇒ accept $\langle w, c \rangle$
- else reject.

- If $w \in L_1 \Rightarrow$ some c makes V accept
- If $w \notin L_1 \Rightarrow$ for all $c \Rightarrow V$ rejects
- V runs in polynomial time