Last time: \( IS, 3\text{-COLOR} \) are \( NP \)-complete.

\[
\text{Clique} = \{ G, k \mid \text{graph } G \text{ has a clique of size } \geq k \}
\]

a clique = subset of vertices st. every pair is connected by an edge

e.g. \[ \square \]

Thm: Clique is \( NP \)-complete.

Proof: (1) Clique \( \in NP \). √
(2) IS \( \leq_P \) Clique

\[
G, k \quad \Rightarrow \quad G', k'
\]

s.t. \( (G, k) \in IS \iff (G', k') \in \text{Clique} \)

IS = \underline{\text{clique}} ...

Define the reduction: \( (G, k) \mapsto (G^c, k) \)

the complement of \( G \)

Vertex Cover \( (VC) = \{ G, k \mid G \text{ has a vertex cover of size } \leq k \} \)

A vertex cover is \( S \subseteq V \)

s.t. \( \forall e = (u,v) \in E \quad (u \in S \lor v \in S) \)

Thm: VC is \( NP \)-complete.

Proof: (1) VC \( \in NP \). √
(2) IS \( \leq_P \) VC
Proof: (1) $VC \in NP$ ✓
(2) $IS \leq_P VC$

$(G, k) \mapsto (G', k')$ s.t.
$(G, k) \in IS \iff (G', k') \in VC$

Claim: $S$ is IS of $G$ $\iff$
$S$ is VC of $G$

Proof: $\Rightarrow$ Let $S$ be an independent set of $G$.
$\forall e = (u, v) \in E$, if $u \in S$ & $v \notin S$,
then $S$ is not independent set.

$\Leftarrow$ Let $S$ be a vertex cover of $G$.
$\forall u, v \in S$, if $(u, v) \in E$,
then $S$ is not a vertex cover.

So the $IS \leq_P VC$ reduction is:

$(G, k) \mapsto (G, n-k)$
where $n = |V| = \# \text{ nodes in } G$

$st$-HamPath $= \{ < G, s, t > | G \text{ has a } st \text{- Hamiltonian path from } s \text{ to } t \}$

A Hamiltonian path is a simple path that visits every node of $G$. 
A Hamiltonian path is a simple path that visits every node of \( G \).

\[ \text{HamCycle} = \{ \langle G \rangle \mid G \text{ has a Ham. cycle} \} \]

A Hamilton cycle is a cycle that passes through every node of \( G \) exactly once.

Thus: \text{st-HamPath} \& \text{HamCycle} are \text{NP-complete}.

**Traveling Salesman Problem (TSP)**

Given a complete graph \( G \) on \( n \) nodes, positive integer weights on edges \( w : E \rightarrow \mathbb{Z}^+ \), and an integer \( W > 0 \), decide if there is a Hamilton cycle in \( G \) with total edge weight \( \leq W \).

The sum of edge weights on the cycle.

TSP is NP-complete.

(\text{HamCycle} \leq_p \text{TSP}. Exercise.)

**Subset Sum**

Given: \( a_1, a_2, \ldots, a_n, \, T \in \mathbb{N} \)

Decide: Is there a subset \( S \subseteq \{1, \ldots, n\} \) such that \( \sum_{i \in S} a_i = T \)?
Thus: Subset Sum is NP-complete.

Proof: (1) Subset Sum ∈ NP
(2) 3SAT ≤ \text{p} Subset Sum
(seethelecturenotesforthedetails.)  \[ \square \]

\[ \text{PSPACE} \]

\[ \text{PSPACE} = \bigcup_{c \geq 0} \text{SPACE}(n^c) \]

\[ \text{space: \# tape cells} \]
\[ \text{touched by a TM} \]
\[ \text{during its run} \]

\[ \text{NPSPACE} = \bigcup_{c \geq 0} \text{NSPACE}(n^c) \]

\[ \text{max space} \leq n^c \]
\[ \text{over alldeterministic} \]
\[ \text{computation branches of} \]
\[ \text{a given NTM.} \]

\[ \text{Thm: NP \leq \text{PSPACE}} \quad \text{(Exercise!)} \]

\[ \text{Thm [Savitch's Theorem]:} \]
\[ \text{NPSPACE = PSPACE.} \]

\text{More precisely,}
\[ \text{NSPACE}(s) \leq \text{SPACE}(s^2) \]
**Proof:**

\[
\text{NTM } M \times 1 \times n
\]

uses space \(\leq S = S(n)\).

Configuration graph:

```
  Cont_0 ----> Cont_1 ----> Cont_2 ----> Cont_3
             |                |
             v                v
             Cont_4
```

1. 0/1 bits (1) state of \(M\)
2. \(O(\log n)\) bits (2) position on input tape
3. \(O(S)\) bits (3) contents of all worktapes
4. \(O(S)\) bits (4) positions on the worktapes

\[2: O(S + \log n) \text{ bits } = O(S)\]

(assume \(S > \log n\))

\[\# \text{ configs } \leq 2^{O(S)}\]

Nodes of Config graph

```
  Cont ----> Cont
  'up'
M in Cont goes to Cont' (legal move)
```

**Our task:** \(O(S)\)
Our task: \( O(s) \)
digraph \( G \) on \( N=2 \) nodes
\( s, t \in \text{nodes of } G \)
init accepting config config

\[ \text{decide: is there } s \rightarrow t \]
in \( G \) ?

BFS: will solve the reachability problem.

time: linear in the size of the input graph

In our case, time: \( O(s) \)

Space is also \( \gtrsim 2^{N} \). Too much!

Main call: Path (init, accept, \( O(s) \))

Algorithm Path \((x, y, i)\) 

\% check if \( x \rightarrow y \) in \( G \) in \( \leq 2^i \) steps

\( i = 0 \) \ then \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \
\begin{align*}
\text{if } i = 0 \text{ then} & \\
\text{Accept } 1 \quad (x = y \text{ or } (x, y) \in E) & \\
\text{endif} \\
\text{for every node } z \in V & \\
\text{if } \text{Path}(x, z, i-1) \quad \text{AND} & \\
\text{Path}(z, y, i-1) & \\
\text{then Accept} & \\
\text{endif} & \\
\text{endfor} & \\
\text{Reject} & \\
\end{align*}

\text{re-use space}

\text{Space}(i) = \text{Space}(i-1) + O(1) \\
\leq O(1|x|^2)

\text{N nodes} \\
|x| \leq \log N

\text{Space}(\log N) \leq O((\log N)^2)

\text{Space}(m) = \text{Space}(m-1) + a \downarrow \\
\text{Space}(mn) = \Omega(m \cdot a)