For some problems it is difficult to find an algorithm in $P$.

We first discuss an example of finding a Hamiltonian Path.

Definition: for a given directed graph find a path going through all nodes once.

Fig 7.17
Hamiltonian Path II

- **HAMPATH** = \{⟨G, s, t⟩ | G : directed, a Hamiltonian path from s to t\}
- A brute-force way: checking all possible paths
  But the number is exponential
- Polynomial verification
  for a path, in P time ⇒ a Hamiltonian path or not
This is an example where verification is easier than determination
We discuss another example where verification is easier than determination.

An integer is composite if

\[ x = pq, \quad p > 1, \quad q > 1 \]

Given \( x \), difficult to find \( p, q \)

Given \( x, p, q \) easily verify \( x = pq \) or not
Some problems are difficult so even a polynomial verifier cannot be easily obtained.

**$\text{HAMPATH}$**: given $\langle G, s, t \rangle$ no Hamiltonian path from $s$ to $t$.

Verification may still be difficult.

Given $s$ and $t$ it seems we still need to check all paths.
Verifier I

Definition: an algorithm $V$ is a verifier of a language $A$ if

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

Example: compositeness. $V$ accepts $\langle w, c \rangle = \langle x, p \rangle$, where $p$ is a divisor

Example: Hamiltonian path. $V$ accepts $\langle w, c \rangle = \langle \langle G, s, t \rangle, \text{ a path from } s \text{ to } t \rangle$

$c$ is called a “certificate”
Verifier II

- Definition: a polynomial verifier if it takes polynomial time of $|w|$
- $A$: polynomially verifiable if $\exists$ a polynomial verifier
- Note that we measure time on $|w|$ without considering $|c|$
- For a polynomial verifier, $|c|$ should be in polynomial of $|w|$
- Otherwise, reading $|c|$ already non-polynomial
NP is a class of languages

Definition: a language $\in \text{NP}$ if it has a polynomial verifier

We will prove that this definition is equivalent to that the language is decided by nondeterministic polynomial TM

This is where the name comes from

Some use this as the definition

Note that for nondeterministic TM the running time is by checking the longest branch
Definition:

$$\text{NTIME}(t(n)) = \{ L \mid L \text{ decided by } O(t(n)) \text{ nondeterministic TM} \}$$

$$\text{NP} = \bigcup_k \text{NTIME}(n^k)$$
A list $p_1 \cdots p_m$ is nondeterministically chosen

For each list:

1. Check repetitions
2. Check $s = p_1; t = p_m$
3. Check that for $i = 1 \ldots m - 1, (p_i, p_{i+1})$ is an edge of $G$

Cost on each list is polynomial:

repetitions: $O(m^2)$

$s = p_1, t = p_m : O(m)$

edge check: $O(m^2)$