# 1 NP-completeness

We mentioned in class that HAMPATH $= \{\langle G, s, t\rangle : G$ has a Hamiltonian path from $s$ to $t\}$ is NP-complete, where $s, t$ are two vertices in a directed graph $G$, and a Hamiltonian path is a path which visits each vertex exactly once. Consider the *undirected* version

$$\text{UHAMPATH} = \{\langle G, s, t\rangle : G \text{ has a Hamiltonian path from } s \text{ to } t,\ G \text{ is undirected}\}.$$

**Exercise.** *Show that UHAMPATH is NP-complete.*

**Solution.**
It is in NP since the Hamiltonian path itself can serve as the certificate.

To show that it is NP-hard, we reduce from HAMPATH. Suppose $G, s, t$ are the inputs for HAMPATH. We transform to an input $G', s', t'$ for UHAMPATH as follows via a Karp reduction. Each vertex $v$ in $G$ other than $s, t$ is transformed into three vertices $v_{in}, v_{mid}, v_{out}$ in $G'$. For $s$, we only create one vertex $s_{out}$ in $G'$, and for $t$ we create $t_{in}$. For every $v \in G \backslash \{s, t\}$, we create edges $(v_{in}, v_{mid})$ and $(v_{mid}, v_{out})$ in $G'$. Also if $(u, v)$ is an edge in $G$, we create the edge $(u_{out}, v_{in})$ in $G'$. Finally, we set $s' = s_{out}$ and $t' = t_{in}$.

We need only show that $G$ has a Hamiltonian path from $s$ to $t$ iff $G'$ has one from $s'$ to $t'$. Suppose $G$ has the desired Hamiltonian path $s = v_0, v_1, v_2, \ldots, v_N = t$. Then in $G'$ we have the Hamiltonian path $s_{out}, (v_1)_{in}, (v_1)_{mid}, (v_1)_{out}, (v_2)_{in}, (v_2)_{mid}, (v_2)_{out}, \ldots, (v_{N-1})_{in}, (v_{N-1})_{mid}, (v_{N-1})_{out}, t_{in}$. Also if we have a Hamiltonian path from $s'$ to $t'$, it must leave $s_{out}$ to some "in" vertex, then go to a "mid", then "out" before going to the next vertex, etc. So it corresponds to a Hamiltonian path in $G$.

# 2 Consequences of P = NP collapse beyond NP

It turns out that if P = NP, there are some problems which we do not believe to even be in NP which will collapse to P as a consequence. One of these is the *circuit minimization* problem. In this problem, we are given some Boolean circuit $C$ on $n$ inputs using NOT, AND, and OR gates, and we would like to decide whether there is a smaller circuit (in terms of number of gates) $C'$ which computes the same exact function on the domain $\{0, 1\}^n$ as $C$.

**Exercise.** *Prove that if P = NP, then circuit minimization is also in P.*

**Solution.**
Let $A(\langle C\rangle, \langle C'\rangle, x)$ be a Boolean which outputs 1 iff the circuit described by $\langle C\rangle$ outputs the same value as the circuit described by $\langle C'\rangle$ on input $x$, and furthermore $C'$ is a smaller circuit than $C$ (else $A$ outputs 0). Then for the circuit minimization language $L$, $\langle C\rangle \in L$ iff $\exists C'\ \forall x\ A(\langle C\rangle, \langle C'\rangle, x) = 1$.

Now, suppose P = NP. Define the language $L' = \{\langle C, C' \rangle : \forall x \ A(\langle C \rangle, \langle C' \rangle, x) = 1\}$. Now notice $L'$ is in co-NP, and co-NP = NP since P = NP. So $L'$ can be decided in polynomial time and thus is decided by some poly-time algorithm $V(\langle C \rangle, \langle C' \rangle)$. Now note $\langle C \rangle \in L$ iff $\exists C' \ \langle C, C' \rangle \in L'$, which is the same as $\exists C' \ V(\langle C \rangle, \langle C' \rangle) = 1$. But this means $L$ has a poly-time verifier $V$, where $\langle C' \rangle$ is serving as the witness, so $L \in$ NP using the verifier definition of NP. Thus $L \in$ P.