Introduction to Algorithms

CS 430

Lecture 25

Outlines

- Announcement
- NPC

Announcement

1. Earn Your Extra Credit: present your project with your teammates
   - 3%
   - when: lecture time on April 25 (Tues.)
   - where:
     - **in-person section:**
       - ❖ SB104
       - ❖ upload your slides to the shared folder ahead of the lecture time at:

         https://drive.google.com/drive/folders/1KO1cuELoq35NMQit_PpRb4y7MrZD_vTG?usp=sharing
     - **online and asynchronous section:**

       https://flip.com/d765fd7a
   - duration: 5 mins as most
   - what to do: presentation should be conducted by ALL team members.
2. 4/28 (Thur.): final review and Q&A

P Problem

- P Problems
  - Theta(n^a), where a is a constant.
  - What if a is large?
    - Is $f(n)=n^{2000}$ practical?
  - Two Supporting Theories
    - more efficient solutions follow the first one;
    - polynomials' properties

- NP Problems
  - Non-deterministic Polynomial
  - Problem *Q* is a NP problem when
    - a **certificate** of a solution can be proved to be true in a polynomial time.
    - ex: satisfiability of a k-cnf

      2-cnf: $(x1 \vee \neg x2) \wedge (x3 \vee \neg x1) \wedge (\neg x2 \vee \neg x3)$

      3-cnf: $(x2 \vee x5 \vee \neg x7) \wedge (x1 \vee \neg x4 \vee \neg x6) \wedge (\neg x1 \vee \neg x3 \vee \neg x6)$

– NP

- NP is the set of decision problems solvable in polynomial time by a non-deterministic Turing machine.
- Two Phases
  - Guess—non-deterministic
  - Verification--deterministic

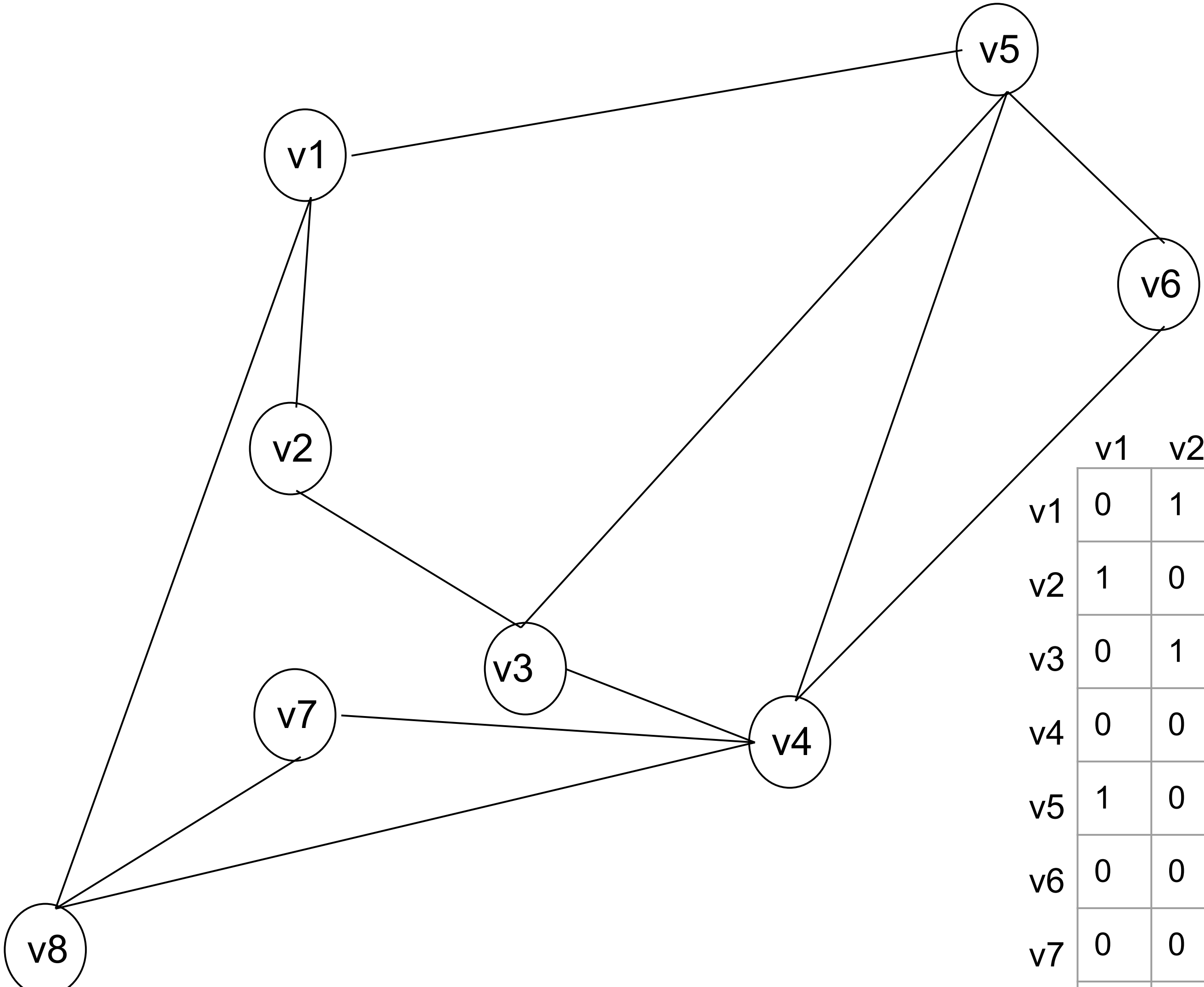
Questions:

- Is a P problem an NP problem?

- $P \subseteq NP$ or $P \subset NP$?

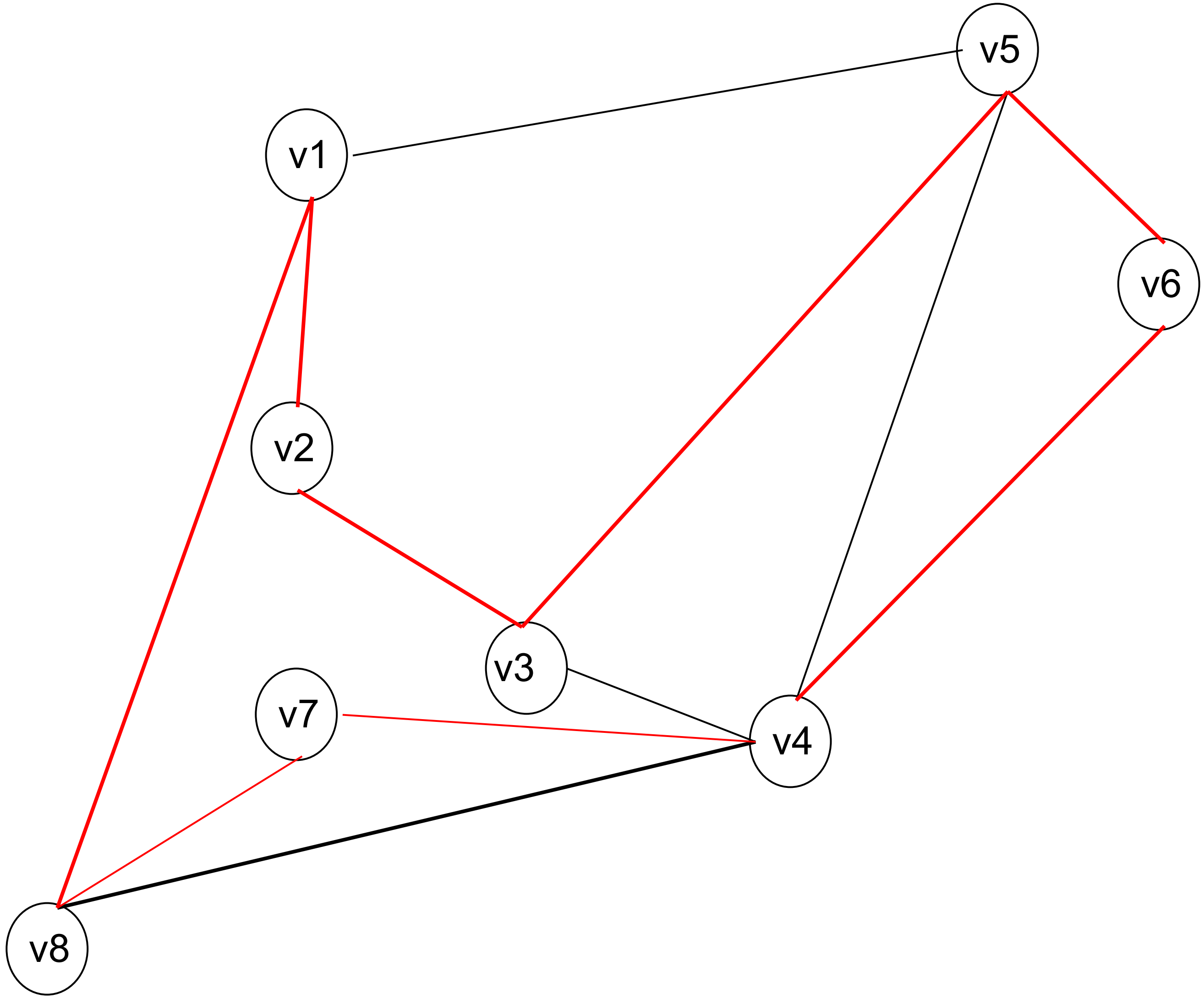
- Hamiltonian Cycles—an example of NP
  - A **single** cycle that contains each vertex once in V in an undirected graph G.
  - Formulation:
    - HAM-CYCLE={<G>: G is a hamiltonian graph}
      - Proof of HAM-CYCLE being an NP
        - » Two phases:
        - » Non-polynomial time to determine a graph being hamiltonian;
        - » Polynomial time to verify a graph to be a hamiltonian when a graph is presented as a hamiltonian.

Ex: G(V,E)



	v1	v2	v3	v4	v5	v6	v7	v8
v1	0	1	0	0	1	0	0	1
v2	1	0	1	0	0	0	0	0
v3	0	1	0	1	1	0	0	0
v4	0	0	1	0	1	1	1	1
v5	1	0	1	1	0	1	0	0
v6	0	0	0	1	1	0	0	0
v7	0	0	0	1	0	0	0	1
v8	1	0	0	1	0	0	1	0

Ex: G(V,E)



$G = (V, E)$

Adjacency Matrix of $G = \begin{array}{c|cccc} & v_1 & v_2 & \cdots & v_m \\ \hline v_1 & 0 & 1 & \cdots & 1 \\ v_2 & 1 & 0 & \cdots & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ v_m & 1 & 1 & \cdots & 0 \end{array}$

The representation of G is a sequence of bits, and the length is $m \times m$. That is $n = m \times m$.

A possible method to determine a solution to be a hamiltonian-cycle is the all permutations of all vertices. That is $m!$.

The complexity is $\Omega(m!) = \Omega(\sqrt{n}!) = \Omega(2^{\sqrt{n}})$

not a polynomial!

- Phase two proof:
  - whether it contains all vertices in V
  - whether all edges between consecutive vertices in the cycle exist in E.
  - $O(n^2)$---polynomial-time

Ex:

A Boolean formula consists of n Boolean variables and m Boolean connectives. For example, $\varphi = ((x_1 \rightarrow x_2) \vee \neg x_3) \wedge \neg x_4$ is a Boolean formula. If there is an assignment of x_i that causes a Boolean formula to 1, this assignment is a *satisfying* assignment and the Boolean formula is ***satisfiable***. To determine a Boolean formula satisfiable is a problem and defined as follows:

L= $\{< \varphi > : \varphi \ is\ a\ satisfiable\ Boolean\ formula\}$

Show and prove that L is NP.

NP Completeness (NPC)

- Problem reducibility
    - Q can be rephrased to Q’ and a solution to Q’ is also a solution to Q:
    - example: a linear equation ax-c=0 can be rephrased to a linear group:

$$\begin{cases} ax+by-c=0 \\ y=0 \end{cases}$$

    - Q is not harder than Q’
    - polynomial time reducible

if Q can be rephrased to Q’ in a polynomial, $Q \le_p Q'$

- An abstract problem L is an NPC when
    - L is an NP, and
    - for any $L' \in NP$, $L' \le p\ L$

- NP - Hard
  - Not necessarily satisfies item 1
  - Satisfies item 2