• You know the drill now: Find students around you to form a *small group*; use *all resources* to help to solve the problems; *discuss* your idea with other group member and *write down* your own solutions; raise your hand and pull the *course staffs* to help; *submit* your writeup through Gradescope in *24 hours*.

Our topic for this working session is on NP-hardness reductions via gadget constructions.

Key takeaway when reducing from 3SAT to your favorite problem R:

- Look for structure in instance of *R* that give you a *choice* for assignments; use them to build *variable gadget*.
- Look for structure in instance of *R* that give you a *interaction with multiple objects*; use them to build *clause gadget*.
- Look for *constraints* in the definition of *R*; use them to model variable assignments (*not*) *satisfying* clauses.

An *induced cycle* in an undirected graph G a subgraph G[S] induced on the vertices in S, that happens to be a cycle. (Intuitively, an induced cycle is a cycle in G without *chords*.)

INDUCEDCYCLE

- Input: An undirected graph G, two vertices u and v
  - **Output:** Is there an induced cycle in G passing through u and v?
- 1. Why doesn't the following construction work?



2. Prove that INDUCEDCYCLE is NP-hard.

To think about later: (No submissions needed)

3. Prove that the following problem is NP-hard:

INDUCED**C**YCLE

- Input: An undirected graph G, one vertex u
- Output: Is there an induced cycle in G of even length passing through u?