CS4234
Optimiz(s)ation Algorithms

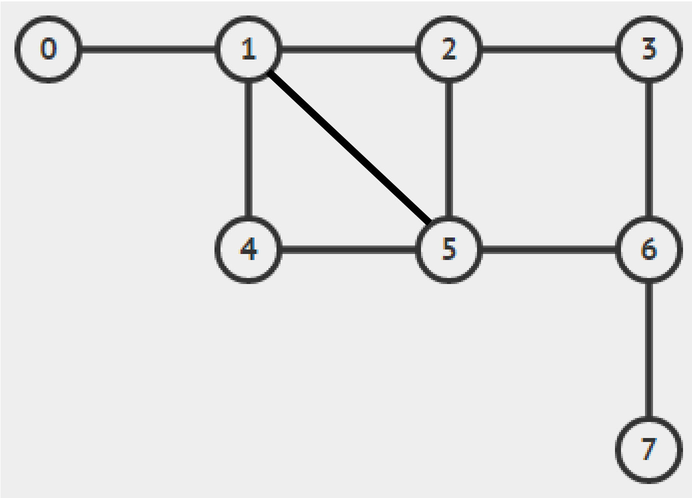
Subset of L1+2 – `Min-Vertex-Cover`

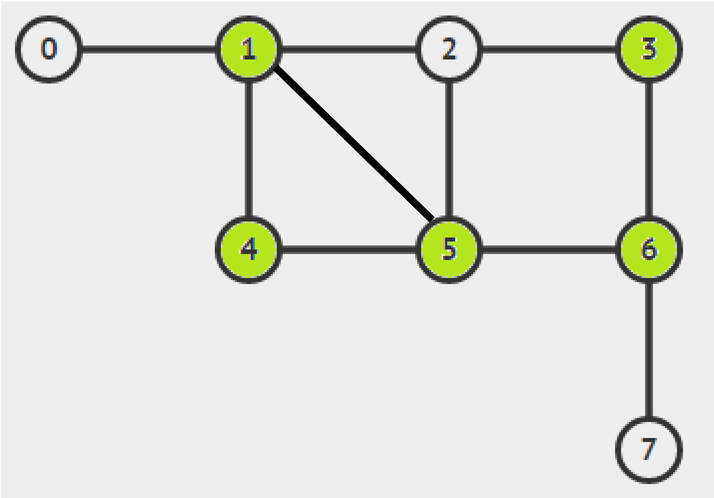
https://visualgo.net/en/mvc

Vertex Cover

Definition:

- A vertex cover for a graph $G = (V, E)$ is a set $S \subseteq V$ such that for every edge $e = (u, v) \in E$, either $u \in S$ or $v \in S$





`Vertex-Cover` is NP-Complete

The **decision** version

- Given a graph G = (V, E) and a parameter k, does there exist a `Vertex-Cover` of G of size k (vertices)?

Proof:

- `Vertex-Cover` is in NP
- `Vertex-Cover` is NP-hard
  - `Clique` $\leq_p$ `Vertex-Cover`

1. **VERTEX-COVER∈NP**

Input: An undirected graph **G = (V, E)** and an integer **k**

Certificate: A subset **V'** of size **k**

The **O(V+E)** verification algorithm checks:

- if **|V'| = k** and insert those vertices into a ___________ (**O(1)** per that data structure insertion, so **O(V)** overall)
- Then, it scans all edge **(u, v)** ∈ **E** to check if at least one of **u** and **v** belongs to **V'** (**O(1)** per that data structure check, so **O(E)** overall)

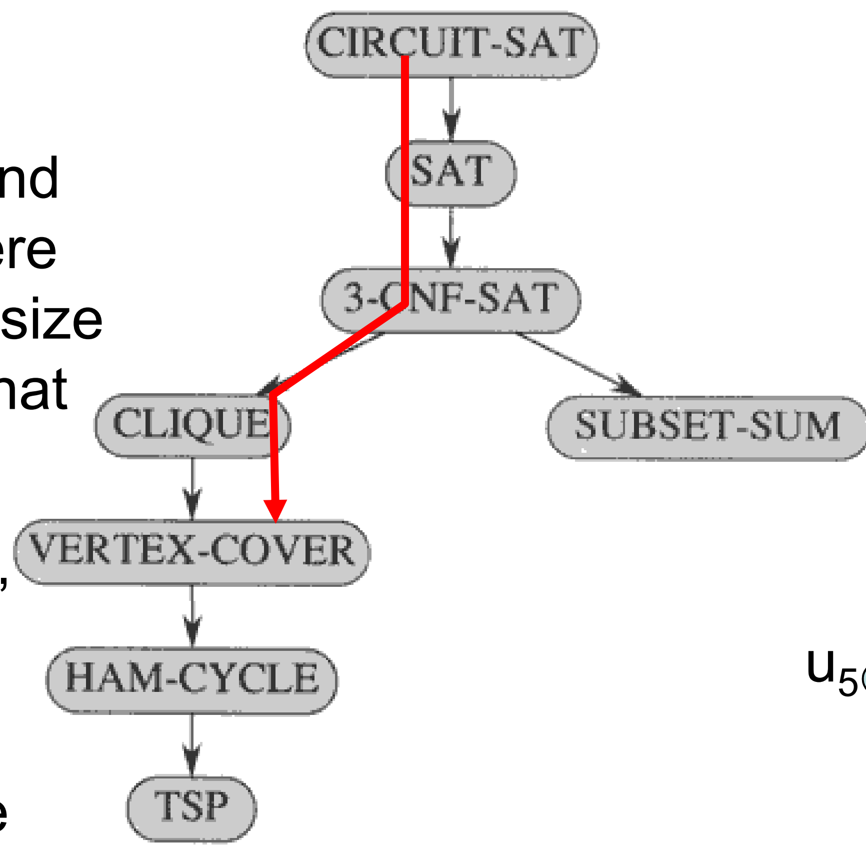
Therefore, **Vertex Cover is in NP**

Agenda for showing more NP-complete problems

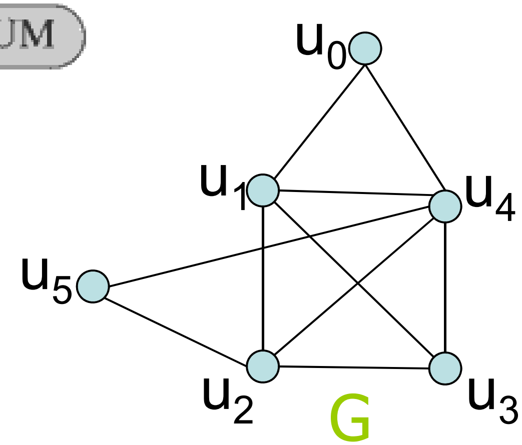
Using polynomial time reduction,
we can obtain **more** NP-complete problems

CLIQUE: Given a graph **G = (V, E)** and an integer **k**, is there a subset **C** ⊆ **V** of size **k** (vertices) such that **C** is a clique in **G**?

For now, "assume" that CLIQUE has been proven to be NP-complete



Eyeball: Is there a clique of size k = 4 below?

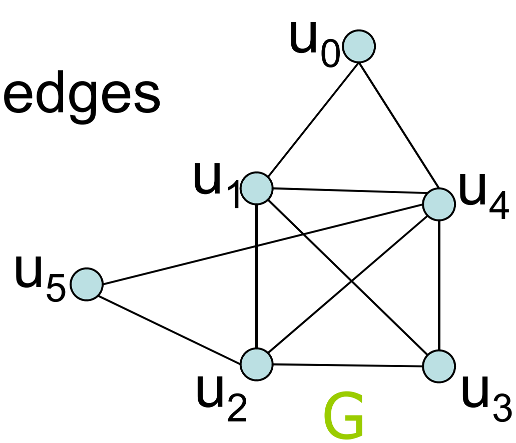


2. CLIQUE$\leq_p$VERTEX-COVER (1)

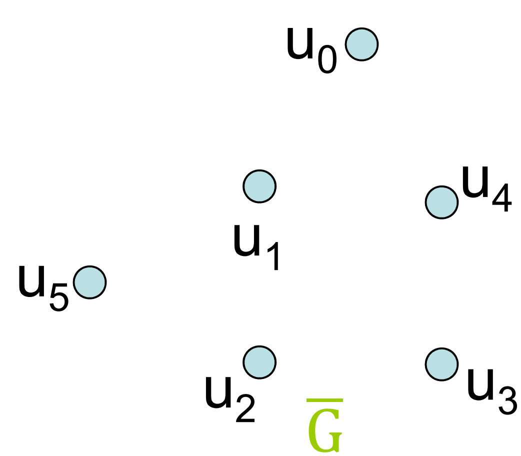
Given an undirected graph **G = (V, E)** and **k** (note: **n = |V|**), we construct a graph $\bar{G}$ **= (V, $\bar{E}$)** where **(u, v)** $\in$ **$\bar{E}$** iff **(u, v)** $\notin$ **E**

Claim: G has a size-**k** clique iff $\bar{G}$ has a size-**(n-k)** vertex cover

Graph **G**
6 vertices,
10 edges
K_6 has 15 edges



Exercise: Draw the complement graph $\bar{G}$ in O(V²), i.e., polynomial time

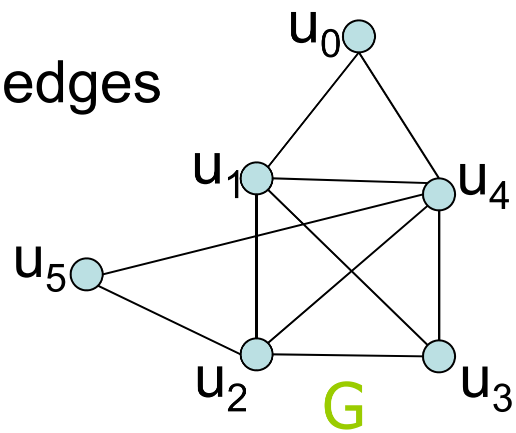


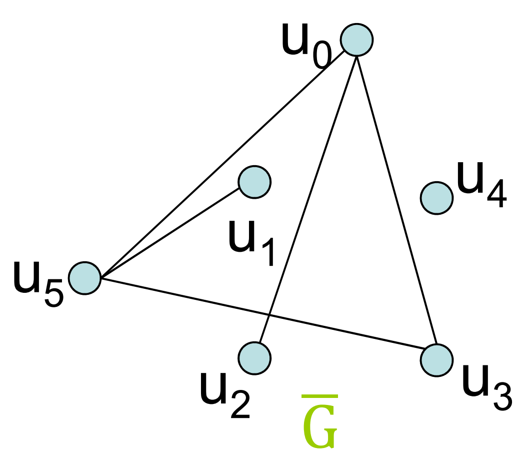
2. CLIQUE$\leq_p$VERTEX-COVER (2)

Given an undirected graph **G = (V, E)** and **k** (note: **n = |V|**), we construct a graph $\bar{G}$ **= (V, $\bar{E}$)** where **(u, v)** $\in$ **$\bar{E}$** iff **(u, v)** $\notin$ **E**

Claim: G has a size-**k** clique iff $\bar{G}$ has a size-**(n-k)** vertex cover

Graph **G**
6 vertices,
10 edges
K_6 has 15 edges





Complement graph $\bar{G}$
6 vertices,
5 edges

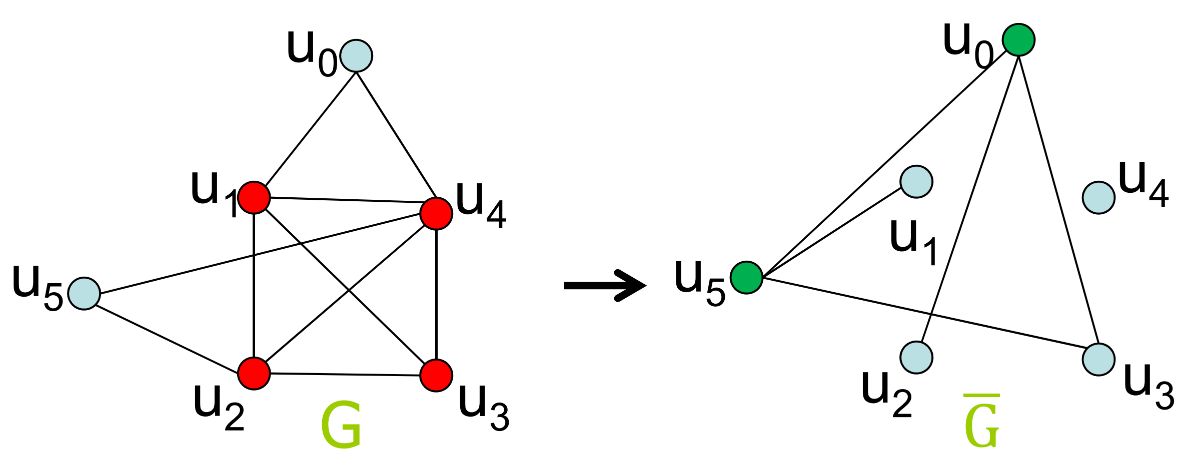
G has a size-**k** clique → $\overline{G}$ has a size-**(n-k)** vertex cover

Suppose V' ⊆ V is a size-**k** clique of G,
e.g., V' = {u_1, u_2, u_3, u_4} is a size-**4** clique of G

- For any arbitrary edge (u, v) ∈ $\bar{E}$ in the complement graph $\overline{G}$, then (u, v) ∉ E
  - Which implies that at least one of u or v does not belong to a clique V' as **every pair** of vertices in V' are connected by an edge in E and thus won't be in $\bar{E}$

Hence, at least one of u and v belongs to V-V' (of size **n-k**),
e.g., V-V' = {u_0, u_5} is a size-**2** vertex cover of $\overline{G}$

- Since edge (u, v) ∈ $\bar{E}$ was chosen arbitrarily, every edge (u, v) ∈ $\bar{E}$ is covered by a vertex in V-V', so V-V' (of size **n-k**) is a VC of $\overline{G}$

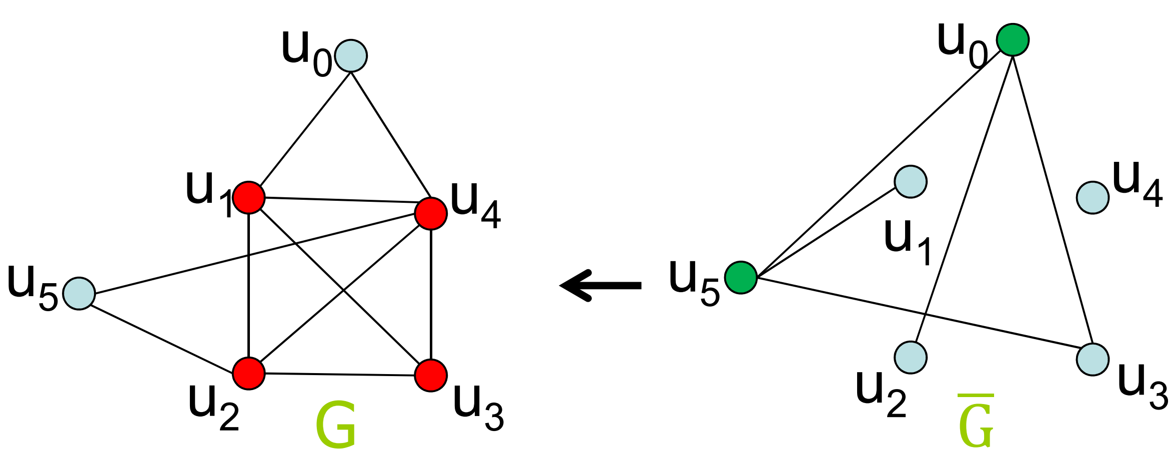


G has a size-**k** clique ← $\overline{G}$ has a size-**(n-k)** vertex cover

Conversely, suppose U ⊆ V is the size-(**n-k**) vertex cover of $\overline{G}$,
e.g., U = {u_0, u_5} is a size-**2** vertex cover of $\overline{G}$

- By definition of vertex cover, for all u, v ∈ V,
  if (u, v) ∈ $\bar{E}$, then at least one of u and v belong to U
- The contrapositive of this statement is for all u, v ∈ V and
  both u and v do not belong to U → (u, v) ∉ $\bar{E}$ → (u, v) ∈ E

Hence, V-U is a clique and V-U has size = **n-(n-k) = k**,
e.g., V-U = {u_1, u_2, u_3, u_4} is a size-**4** clique of G



VERTEX-COVER is NP-Complete

We have shown that:

1. VERTEX-COVER is in NP
2. VERTEX-COVER is NP-Hard

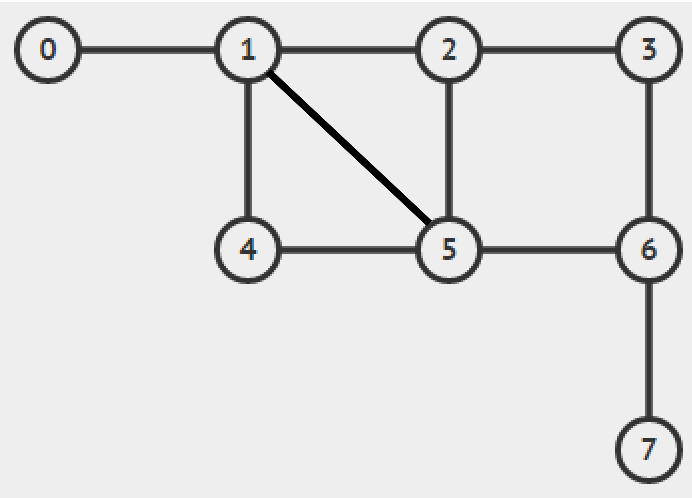
Therefore, VERTEX-COVER (**the decision problem**) is NP-Complete

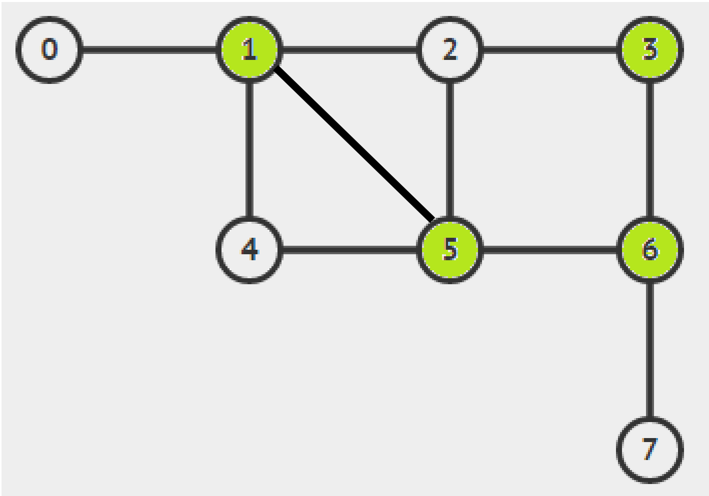
Min-Vertex-Cover (MVC)

Definition:

– Given a graph G = (V, E), find a **minimum-sized** set S that is a vertex cover for G

Try Brute Force ("Complete Search") live at https://visualgo.net/en/mvc

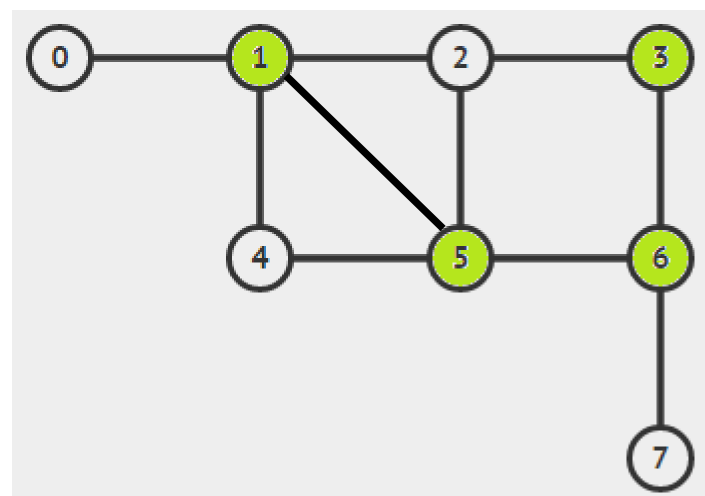




MIN-VERTEX-COVER is NP-hard (1)

Clearly, **if** we could efficiently find a MIN-VERTEX-COVER (MVC, **an optimization problem**), then we could also efficiently answer the **decision version** of VERTEX-COVER (VC), i.e., VERTEX-COVER $\leq_p$ MIN-VERTEX-COVER

Decide: Can we have VC with **k** vertices of this graph?
➔ Just run MVC optimization algorithm on that graph output yes if the ans ≤ **k** or no otherwise



- An **O(1)** (polynomial) reduction (it does nothing actually)

MIN-VERTEX-COVER is NP-hard (2)

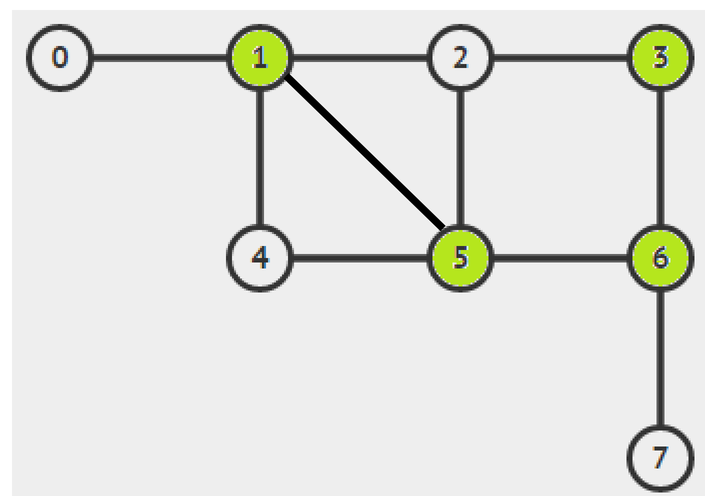
Hence finding a MIN-VERTEX-COVER (MVC) is at least as hard as the decision version, and hence we term it **NP-hard**

But MVC is **not NP-complete**

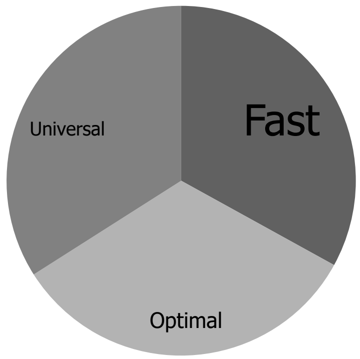
- We do not have polynomial time verifier to show that MVC is in NP

Bonus:

- Solving MVC also solves another NP-hard problem (details in tutorial): Max-Independent-Set (MIS)
  - MIS: un-selected vertices in the example



Dealing with an NP-hard Problem



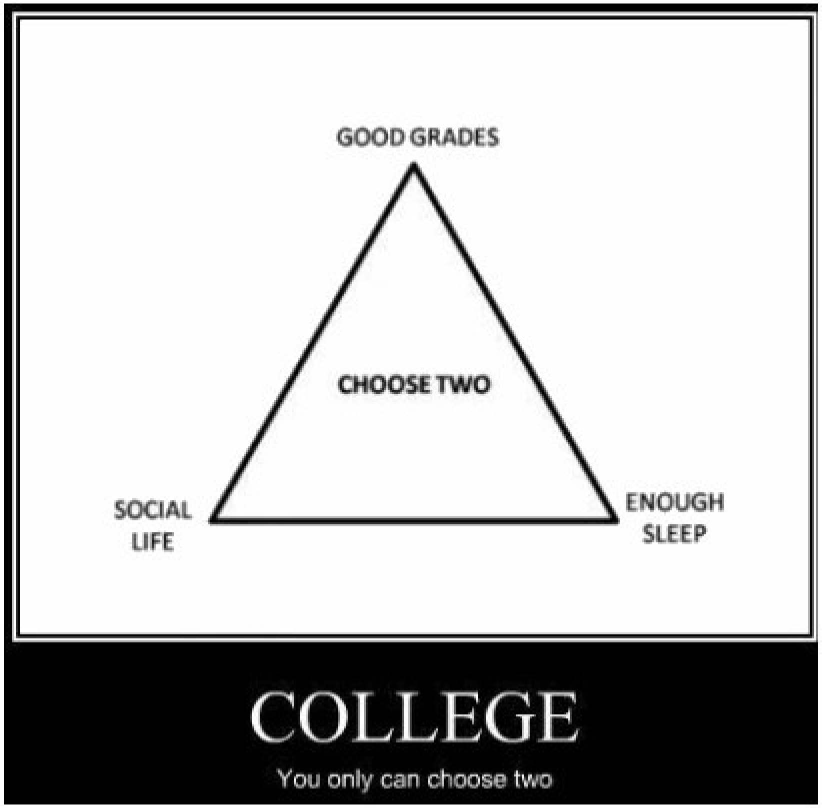
Desirable Solution:

1. Fast (i.e., polynomial time)
2. Optimal (i.e., yielding the best solution possible)
3. Universal (i.e., good for all instances/inputs)

In reality: Choose 2 out of 3:

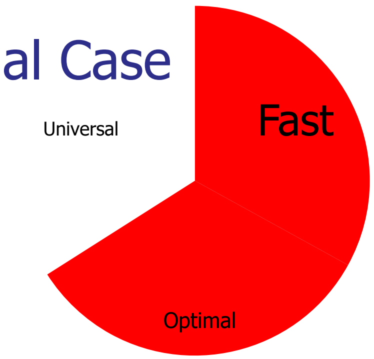
1. Deal with the special case (lost universality)
2. Deal with parameterized solution (lost speed)
3. Consider approximate solution (lost optimality)

Intermezzo



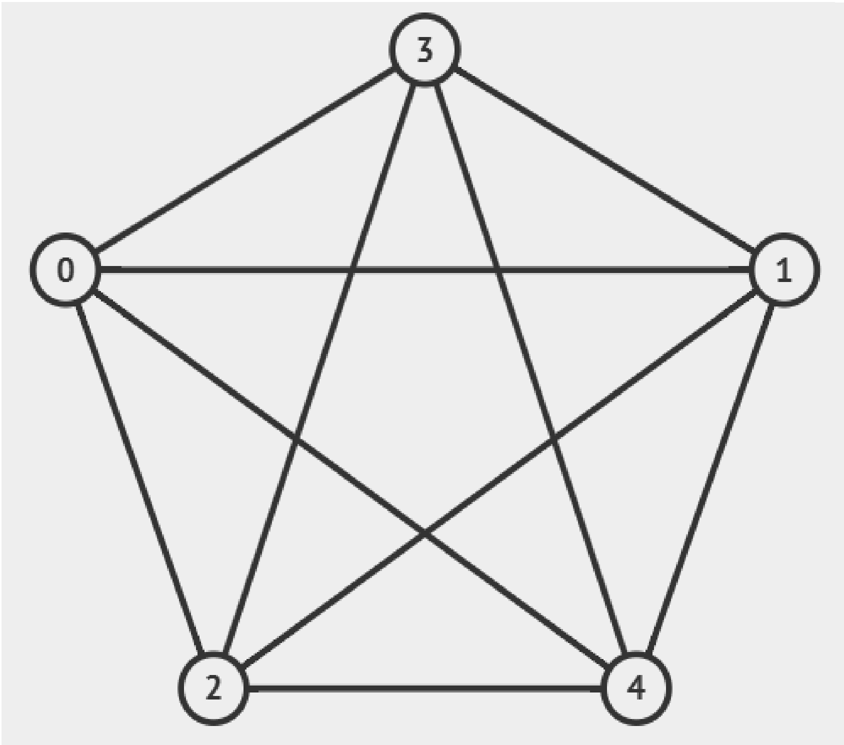
PS: Similar pictures can be easily Googled online

MVC on Complete Graph – Special Case



Trivial $\Theta(1)$ solution to output MVC, $\Theta(V)$ for cert:

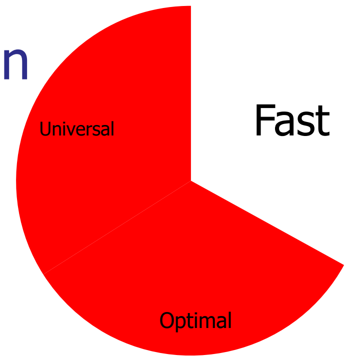
- MVC = |V|-1, certificate: Just select any V-1 vertices
  - The last vertex does not need to be taken into the MVC



PS: There are a few other known special cases of MVC ☺, take CS4234 ☺

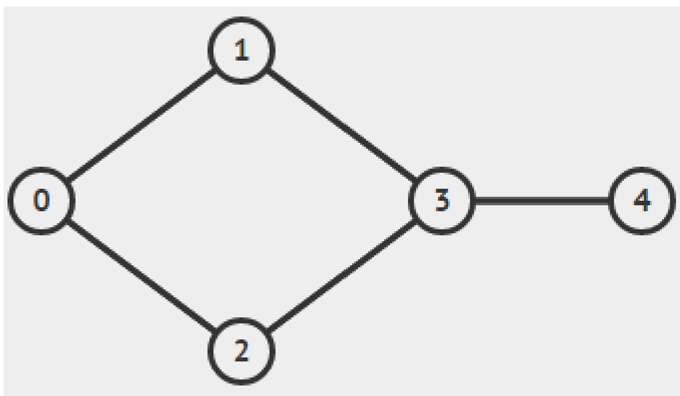
Try live at https://visualgo.net/en/mvc, Example Graphs, Max Clique, K5

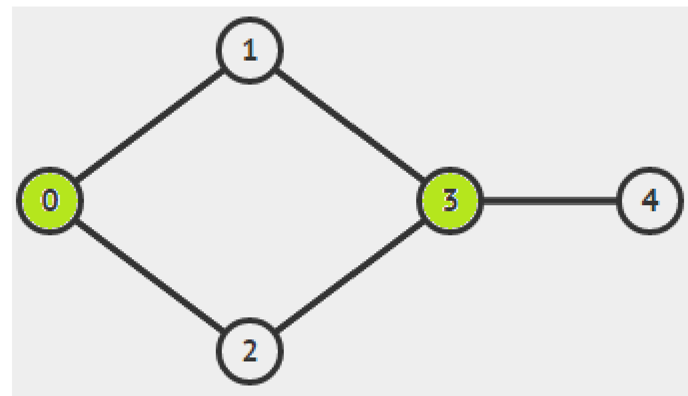
MVC with small k – Parameterized Solution



Parameterized Complexity

– What if you are told that $k \leq 2$?

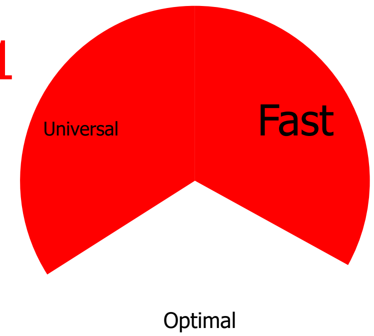




– PS: Pure Brute Force is $O(2^n m)$,
see https://visualgo.net/en/mvc

– Can you come up with an $\Theta(n^2)$ solution?
  - PS: $k \leq 2$ means that k can also be … ?

Approx Algo for MVC – Deterministic 1




```
/* This algorithm adds vertices greedily, one at a time, until everything
   is covered. The edges are considered in an arbitrary order, and for
   each edge, an arbitrary endpoint is added.                          */
```

Algorithm: ApproxVertexCover-1($G = (V, E)$)

Procedure:

$C \leftarrow \emptyset$

/* Repeat until every edge is covered: */

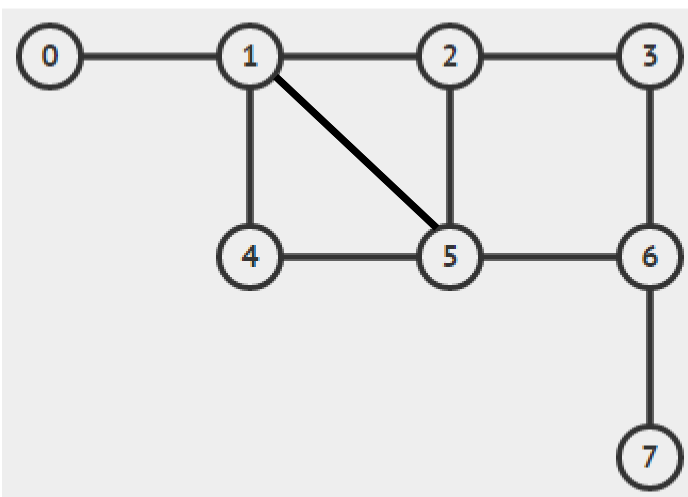
while $E \neq \emptyset$ **do**

  Let $e = (u, v)$ be any edge in G.

  $C \leftarrow C \cup \{u\}$

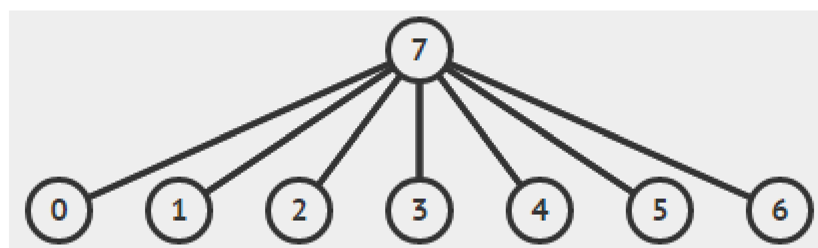
  $G \leftarrow G_{-u}$ // Remove u and all adjacent edges from G.

return C

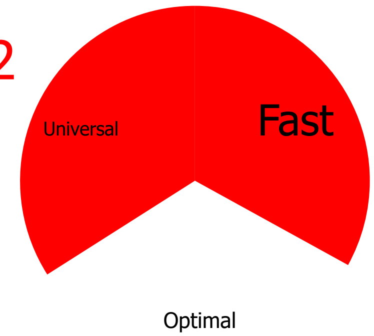


Can be very bad:

– **??**-approximation **(no bound)**



Approx Algo for MVC – Deterministic 2



/* This algorithm adds vertices greedily, two at a time, until everything is covered. The edges are considered in an arbitrary order, and for each edge, both endpoints are added. */

Algorithm: ApproxVertexCover-2($G = (V, E)$)

Procedure:

  $C \leftarrow \emptyset$

  /* Repeat until every edge is covered: */

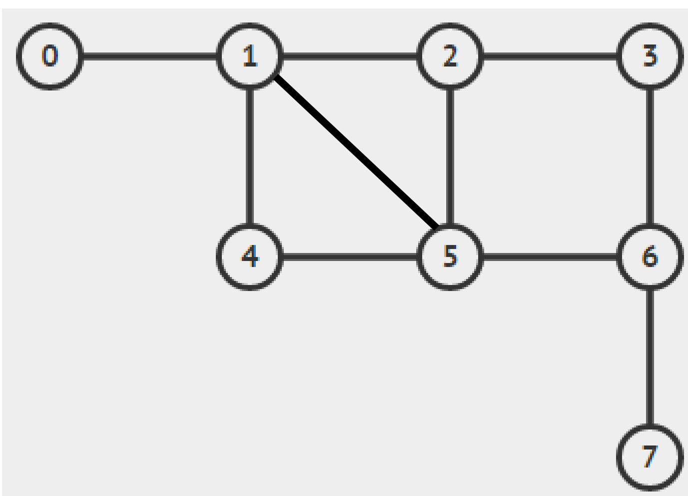
  while $E \neq \emptyset$ **do**

    Let $e = (u, v)$ be any edge in G.

    $C \leftarrow C \cup \{u, v\}$

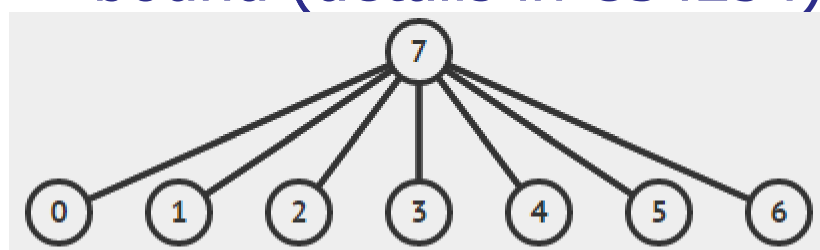
    $G \leftarrow G_{-\{u,v\}}$ // Remove u and v and all adjacent edges from G.

  return C

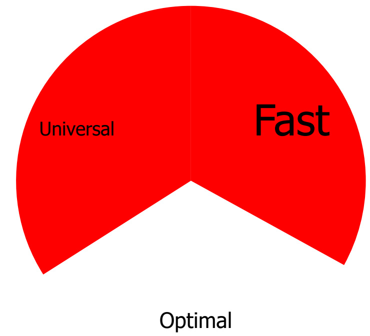


Not that bad?

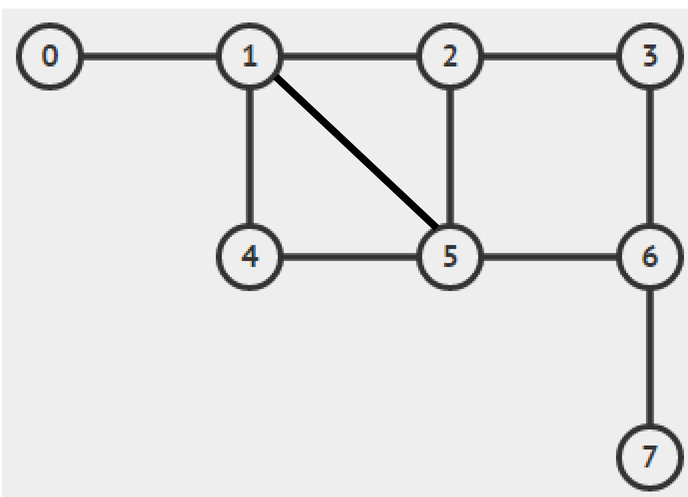
– This one has 2-approximation bound (details in CS4234)



Approx Algo for MVC - Randomized




```
1  Algorithm: RandomizedVertexCover(G = (V, E))
2  Procedure:
3      C ← ∅
       /* Repeat until every edge is covered:                    */
4      while E ≠ ∅ do
5          Let e = (u, v) be any edge in G.
6          b ← Random(0, 1) // Returns {0, 1} each with probability 1/2.
7          if b = 0 then z = u
8          else if b = 1 then z = v
9          C ← C ∪ {z}
10         G ← G_{-z} // Remove z and all adjacent edges from G.
11     return C
```



2-approximation Analysis:

– Watered down proof:

- In line [5-8], the probability that we choose the right z is at least ½
- We include z in the cover and remove z (and 'cover' its edges) in line 9-10
- So C is at most 2 * OPT when E = ∅
  – In expectation *verbal discussion*

Try Probabilistic 2-Opt live at https://visualgo.net/en/mvc

Summary

- VC (decision) is NP-complete
- Introducing the MVC problem, it is NP-hard
- ${}_3C_2$ scenarios
- Special case of MVC: On Complete Graph
- Parameterized MVC: Small $k \leq 2$, good BF
- Approximation algorithms for MVC: Det+Rand