CS340 - Analysis of Algorithms

NP Completeness

Announcements:

HW9 - Due Monday (December 8th)

Submit notes from Lab 9 before Lab tomorrow

Final Exam Options:

1. Dec 12 1-4pm Park 230
2. Dec 15 9:30-12:30 Park 159

Warmup:

1. SAT or UNSAT? $(x1 \vee \neg x2 \vee x3) \wedge (\neg x1 \vee x3 \vee x4)$

2. SAT or UNSAT? $(A \vee B) \wedge (\neg A \vee C) \wedge (\neg B \vee \neg C)$

3. True or False: If OpenAI develops a (provably correct) $O(n^3)$ SAT solver, P = NP

4. True or False: If a problem is NP-hard, then it is in NP

5. True or False: If a problem A is NP-Hard and $A \in P$, then P = NP

Agenda

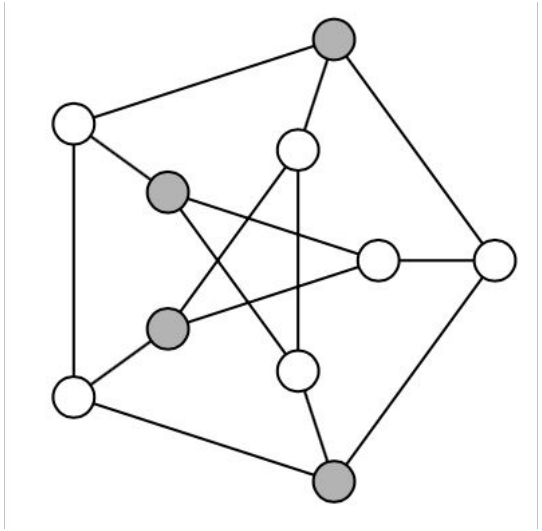
1. An NPC reduction (3SAT to Independent Set)
2. A tour of NPC problems
    a. Vertex Cover
    b. Set cover
    c. Hamiltonian Cycle
    d. Traveling Salesman Problem
    e. Load balancing
3. Worksheet

Independent Set

Given an undirected graph G = (V,E), and an integer k, does G contain a subset V’ of k vertices so that no two vertices in V’ are adjacent to one another?

(G, 4) ?
YES

(G, 5) ?
NO



Show that Independent Set is NPC

Polynomial Time Reduction

If $A \leq_p B$, “A is polynomial-time reducible to B”

A can be solved with polynomial time calls to a black box that solves B

B is at least as hard as A
A is no harder than B

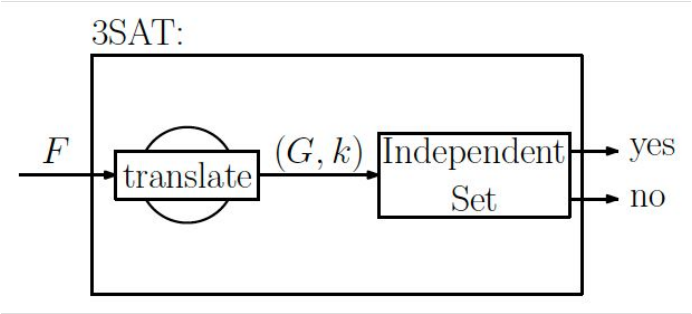
- Reduce to the problem you want to show is harder / not easier
- **If you want to show B ∈ NPC, reduce an A ∈ NPC problem to B ($A \leq_p B$) and show B∈ NP**
- Requirements for a problem *p* to be ∈ NPC:
  - Decision problem
  - Verifiable in polynomial time (∈ NP)
  - For all *j* ∈ NP, j is reducible to *p* ($j \leq_p p$)

Show that Independent Set (IS) is NP-Complete

We know 3SAT is NPC.

To show IS is NPC we will

1. **Show IS ∈ NP**
2. Show IS is NPC by showing 3SAT $\leq_p$ IS (3SAT is reducible to IS)
   a. IS ∈ NPC transitively (j ∈ NP reduces to 3SAT and 3SAT reduces to IS)



Show IS ∈ NP

```
verifyIndependentSet(G, k, V'):
  // 1. Check if V' is a subset of V
  for each v ∈ V':
    if v ∉ V:
      return false // V' contains a vertex not in G

  // 2. Check if the size of V' is at least k
  if |V'| < k:
    return false

  // 3. Check if V' is an independent set (no two vertices in V' are connected by an edge)
  for each u IN V':
    for each v IN V':
      if u != v && (u, v) ∈ E:
        return false

  return true
```

Runtime?

$O(n^2)$

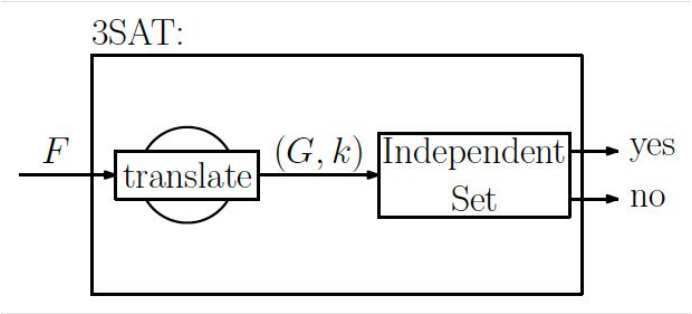
So, **IS ∈ NP**

Show that Independent Set (IS) is NP-Complete

We know 3SAT is NPC.

To show IS is NPC we will

1. Show IS ∈ NP
2. **Show IS is NPC by showing 3SAT $\leq_p$ IS (3SAT is reducible to IS)**
   a. IS ∈ NPC transitively (j ∈ NP reduces to 3SAT and 3SAT reduces to IS)



3SAT $\leq_p$ IS Reduction Design

Given a 3SAT CNF formula, we need to transform this to an instance of Independent Set

- For 3SAT, we need at least 1 variable in each clause to be true (k vars in k clauses)
  - but a literal and its negation cannot both be set to true

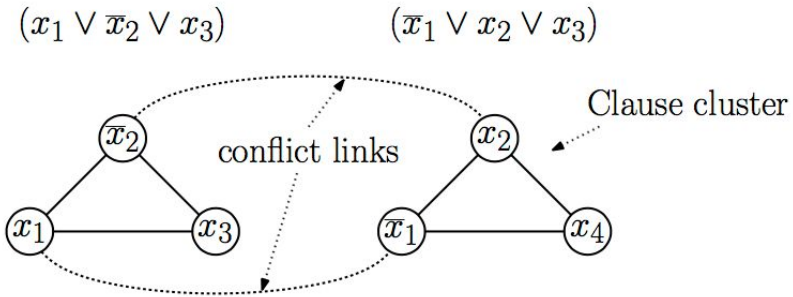
- For Independent Set, we need at least k vertices in V’
  - but none of the vertices in V’ can have an edge between them

3SAT $\leq_p$ IS Reduction Design

Given a 3SAT CNF formula, we will construct a G=(V,E) where

V = create a vertex for each instance of a literal

E = connect vertices with their negations and with other literals in the clause



3SAT $\leq_p$ IS Reduction

$$F = (x_1 \vee \overline{x_2} \vee \overline{x_3}) \wedge (\overline{x_1} \vee x_2 \vee x_3) \wedge (\overline{x_1} \vee x_2 \vee \overline{x_3}) \wedge (x_1 \vee \overline{x_2} \vee x_3)$$

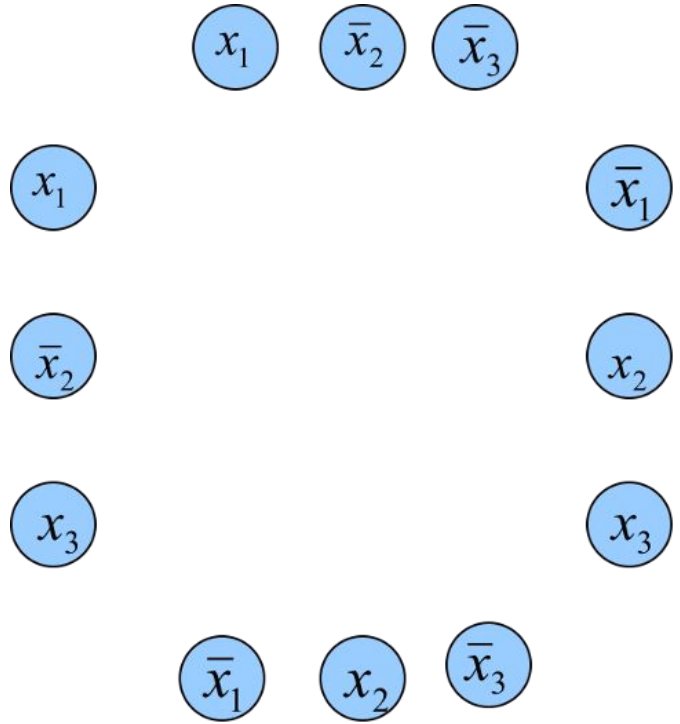
Given a 3SAT CNF formula F,

```
k = number of clauses in F
//create G
for each (clause (x1 or x2 or x3) in F)
  create a clause cluster with vertices x1, x2, x3
  create edges (x1, x2), (x2, x3), (x3, x1)
for each (vertex xi)
  // conflict edges
  create edges between xi and all its complement
return (G, k)
```

Runtime complexity of the reduction?
$3k + (3k)^2$
$O(k^2)$ = polynomial
$1 \leq n \leq 3k$

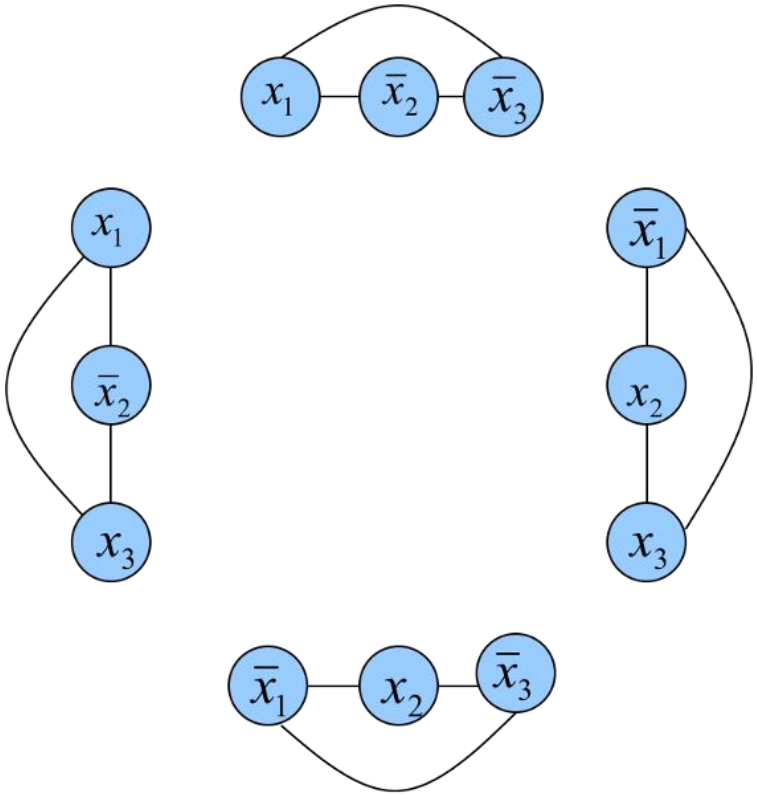
3SAT $\leq_p$ IS Reduction Example

$F = (x_1 \vee \overline{x_2} \vee \overline{x_3}) \wedge (\overline{x_1} \vee x_2 \vee x_3) \wedge (\overline{x_1} \vee x_2 \vee \overline{x_3}) \wedge (x_1 \vee \overline{x_2} \vee x_3)$



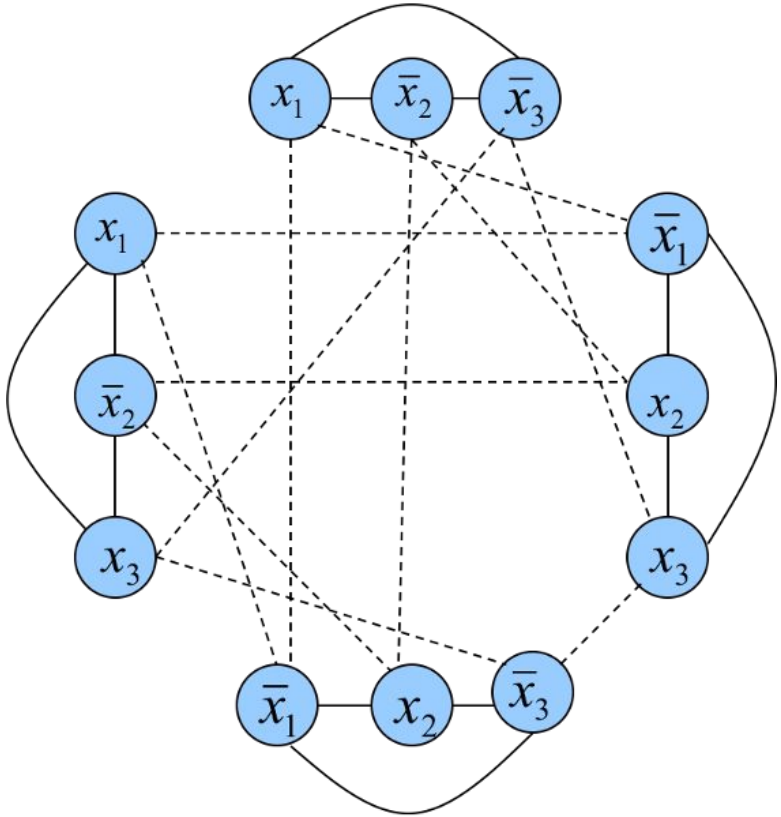
3SAT $\leq_p$ IS Reduction Example

$F = (x_1 \vee \overline{x_2} \vee \overline{x_3}) \wedge (\overline{x_1} \vee x_2 \vee x_3) \wedge (\overline{x_1} \vee x_2 \vee \overline{x_3}) \wedge (x_1 \vee \overline{x_2} \vee x_3)$

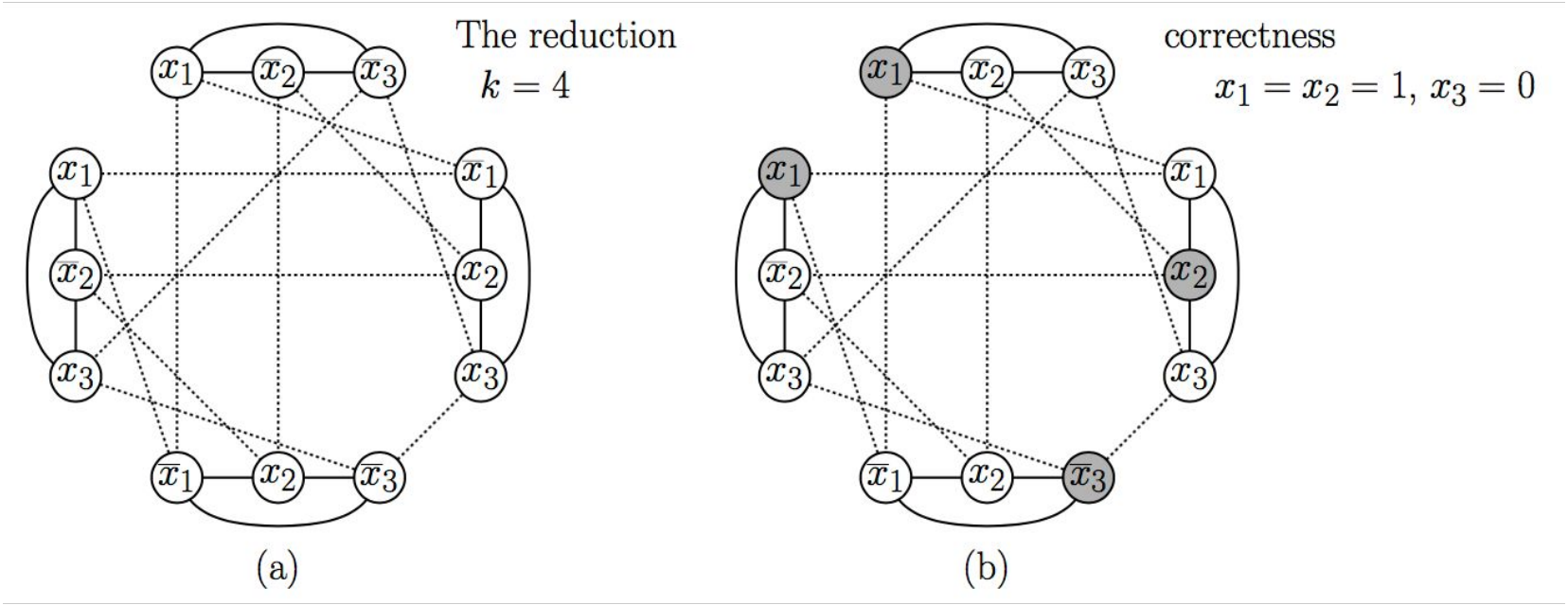


3SAT $\leq_p$ IS Reduction Example

$F = (x_1 \vee \overline{x_2} \vee \overline{x_3}) \wedge (\overline{x_1} \vee x_2 \vee x_3) \wedge (\overline{x_1} \vee x_2 \vee \overline{x_3}) \wedge (x_1 \vee \overline{x_2} \vee x_3)$



3SAT $\leq_p$ IS Reduction Example



Correctness of the Reduction

Claim: F is satisfiable iff G has an independent set of size k

Direction 1: If F is satisfiable, then G has an independent set of size k

- Since F is SAT, each of the clauses must have at least one true literal
- Select one true literal from each clause
- Let V' be the the vertices that correspond to the selected literals
- V' is a valid Independent Set of size k because
  - One literal/vertex was selected from each clause and there are k clauses
  - for all pairs of vertices (u,v), (u,v)$\notin$E b since edges exist between:
    - variables in same cluster (only selected one per cluster)
    - literals and their negations (two conflicting literals cannot be in the interpretation of F)

Correctness of the Reduction

Claim: F is satisfiable iff G has an independent set of size k

Direction 2: If G has an independent set of size k, then F is satisfiable

Map each vertex in V’ to a true variable in F.

Since we put edges between vertices in each cluster, a max of 1 variable from each cluster was selected.

Since |V’| = k, it contains exactly one vertex from each cluster.

We can’t possibly select a literal and its negation since there's edges between them

If we have 1 from each cluster and they’re all valid, it is a satisfying assignment

Full proof posted on webpage

NPC Reduction general advice

Everything is boolean satisfiability!

- There is something the be selected
- Requirement: a sufficient number of such things must be selected
- Restriction: something is limiting our ability to select these things

A reduction needs to determine how to map these elements to each other

Design your reduction without attempting to solve the problem! Assume a black box

Do not treat certain elements differently based on your insights (greedy kills!)

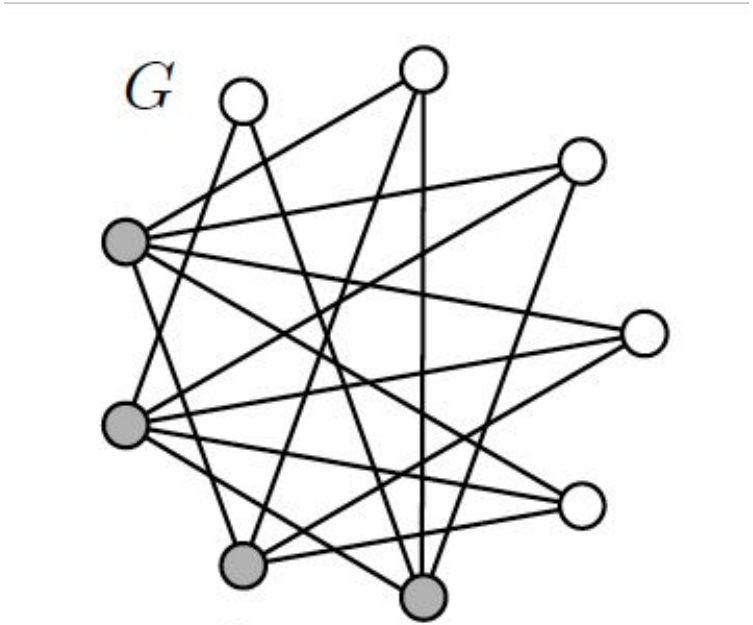
Agenda

1. An NPC reduction (3SAT to Independent Set)
2. **A tour of NPC problems**
    a. Vertex Cover
    b. Set cover
    c. Hamiltonian Cycle
    d. Traveling Salesman Problem
    e. Load Balancing

Vertex Cover (VC)

- For a graph G = (V,E), a vertex cover is a subset V’ ⊆ V such that for every edge is connected to at least one vertex in V’

- Given an undirected graph G = (V, E), and an integer k, does G have a vertex cover of size k?



Set Cover

- Given a set U of n elements, and a collection { S_1 , ... , S_m } of subsets of U, does there exist a collection of at most k of those sets whose union is equal to all of U?
- Example 1: U = {1, 2, 3, 4, 5}
  - S_1 = {1, 2, 3}
  - S_2 = {2, 4}
  - S_3 = {3, 4, 5}
  - S_4 = {4, 5}

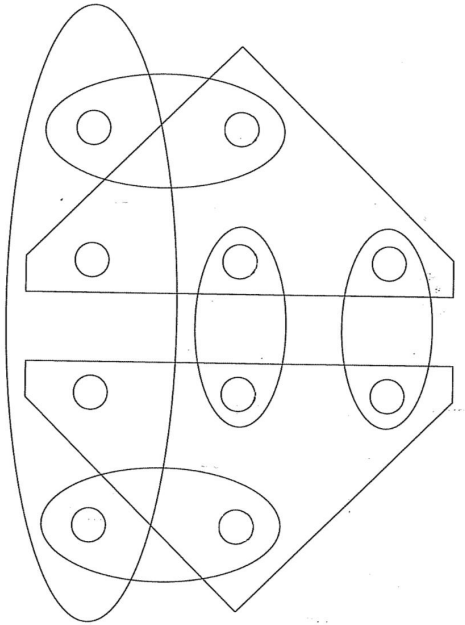
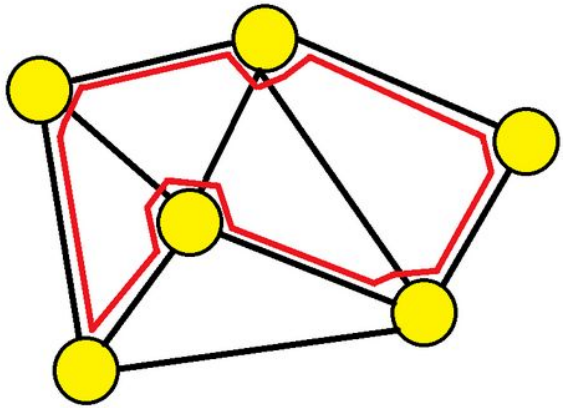


Figure 8.2 An instance of the Set Cover Problem.

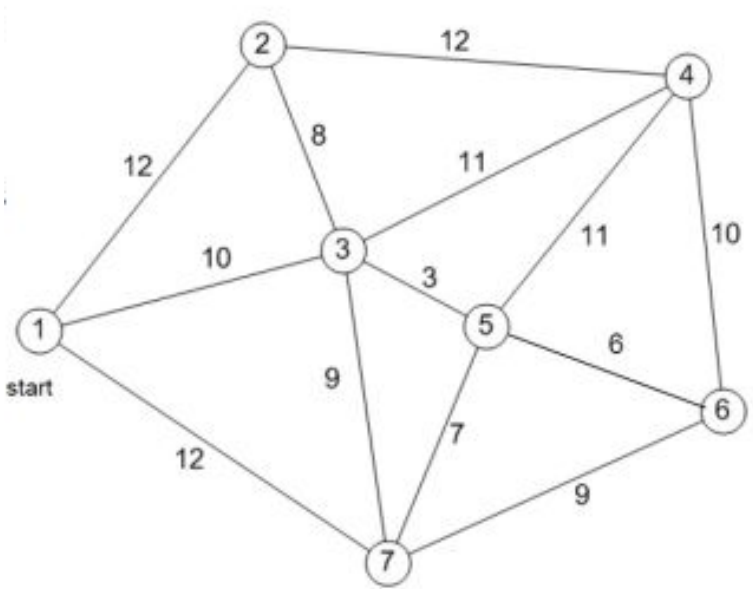
Hamiltonian Cycle

- Given an undirected graph G = (V, E), does there exist a cycle that visits every vertex of G exactly once?



Traveling Salesman Problem (TSP)

Given a complete undirected graph with nonnegative edge weights, find a cycle that visits all vertices in at most k cost



Load Balancing

- Given m identical machines and n jobs, each job j has processing time t_j
  - job j must run contiguously on one machine
  - a machine can process at most one job at a time
- Let $J(i)$ be the subset of jobs assigned to machine i. The *load* of machine i is

$$L_i = \sum_{j \in J(i)} t_j$$

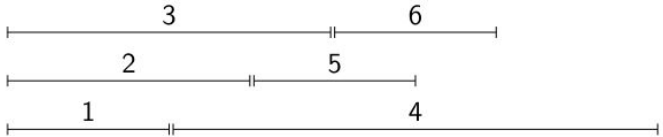
The *makespan* is the maximum load on any machine

$$\max_{i=1,..,m} L_i$$

Goal: Can we find a schedule with a *makespan* of at most k?

Jobs	1	2	3	4	5	6
t_i	2	3	4	6	2	2

Consider the following schedule on 3 machines



List of NP Complete problems

- 3-Colorability (3Col)
- Clique Cover (CCov)
- Independent Set (IS)
- Vertex Cover (VC)
- Hamiltonian Cycle (HC)
- Traveling Salesman Problem (TSP)
- Set Cover
- Load Balancing
- SAT / 3SAT

Worksheet

Summary

- Today we did a reduction from 3SAT to IS
  - Create a vertex for each variable in each clause
  - Add an edge between vertices in the same clause
  - Add an edge between literals and their negations
- Tour of NPC problems
- **HW9 due Monday**
- **Submit Lab 9 notes before lab tomorrow**