Admin

Assignment 11 out today (last one!)

Review next Tuesday

NP problems

NP is the set of problems that can be *verified* in polynomial time

A problem can be verified in polynomial time if you can check that a given solution is correct in polynomial time

(NP is an abbreviation for non-deterministic polynomial time)

P problems

P = problems with a polynomial runtime solution

Also, called “tractable” problems

(Basically, all of the problems in this class)
P and NP

Big-O allowed us to group algorithms by run-time.

Today, we're talking about sets of problems grouped by how easy they are to solve.

P = problems with a polynomial runtime solution (tractable)

NP = problems where a solution is easy to check but finding a solution is not easy.

Reduction function

Given two problems $P_1$ and $P_2$, a reduction function, $f(x)$, is a function that transforms a problem instance $x$ of type $P_1$ to a problem instance of type $P_2$ such that a solution to $x$ exists for $P_1$ iff a solution for $f(x)$ exists for $P_2$.

Proving NP-completeness

Given a problem NEW to show it is NP-Complete:

1. Show that NEW is in NP
   - Provide a verifier
   - Show that the verifier runs in polynomial time
2. Show that NEW is NP-Hard (i.e., all NP-complete problems are reducible to NEW in polynomial time)
   - Describe a reduction function $f$ from a known NP-Complete problem to NEW
   - Show that $f$ runs in polynomial time
   - Show that a solution exists to the NP-Complete problem iff a solution exists to the NEW problem generated by $f$. 
Proving NP-completeness

Show that a solution exists to the NP-Complete problem IFF a solution exists to the NEW problem generated by f.

- Assume we have an NP-Complete problem instance that has a solution, show that the NEW problem instance generated by f has a solution.
- Assume we have a problem instance of NEW generated by f that has a solution, show that we can derive a solution to the NP-Complete problem instance.

Other ways of proving the IFF, but this is often the easiest.

NP-complete: 3-SAT

A boolean formula is in n-conjunctive normal form (n-CNF) if:
- It is expressed as an AND of clauses.
- Where each clause is an OR of no more than n variables.

\[(a \lor \neg a \lor \neg b) \land (c \lor b \lor d) \land (\neg a \lor \neg c \lor \neg d)\]

3-SAT: Given a 3-CNF boolean formula, is it satisfiable?

3-SAT is an NP-complete problem.

NP-complete: SAT

Given a boolean formula of n boolean variables joined by m connectives (AND, OR or NOT) is there a setting of the variables such that the boolean formula evaluate to true?

\[(a \land b) \lor (\neg a \land \neg b)\]

\[((\neg(b \lor \neg c) \land a) \lor (a \land b \land c)) \land c \land \neg b\]

Is SAT an NP-complete problem?

11
1. Show that SAT is in NP
   a. Provide a verifier
   b. Show that the verifier runs in polynomial time

Verifier: A solution consists of an assignment of the variables
- If clause is a single variable:
  • return the value of the variable
- otherwise
  • for each clause:
    • call the verifier recursively
    • compute a running solution
- polynomial run-time?

2. Show that all NP-complete problems are reducible to SAT in polynomial time
   a. Describe a reduction function f from a known NP-complete problem to SAT
   b. Show that f runs in polynomial time
   c. Show that a solution exists to the NP-complete problem IFF a solution exists to the SAT problem generated by f

Reduce 3-SAT to SAT:
- Given an instance of 3-SAT, turn it into an instance of SAT

Reduction function
- DONE
- Runs in constant time (or linear if you have to copy the problem)

3. Show that a solution exists to the NP-complete problem IFF a solution exists to the original problem
   a. Assume we have an NP-complete problem instance that has a solution, show that the original problem instance generated by f has a solution
   b. Assume we have a problem instance of the original problem, show that we can derive a solution to the NP-complete problem instance
   - Assume we have a 3-SAT problem with a solution
     - Because 3-SAT problems are a subset of SAT problems, then the SAT problem will also have a solution
     - Our reduction function simply does a copy, so it is already a 3-SAT problem
     - Therefore the variable assignment found by our SAT-solver will also be a solution to the original 3-SAT problem
CLIQUE

A clique in an undirected graph $G = (V, E)$ is a subset $V' \subseteq V$ of vertices that are fully connected, i.e., every vertex in $V'$ is connected to every other vertex in $V'$.

CLIQUE problem: Does $G$ contain a clique of size $k$?

Is there a clique of size 4 in this graph?

HALF-CLIQUE

Given a graph $G$, does the graph contain a clique containing exactly half the vertices?

Is HALF-CLIQUE an NP-complete problem?

Is Half-Clique NP-Complete?

1. Show that NEW is in NP
   a. Provide a verifier
   b. Show that the verifier runs in polynomial time
2. Show that all NP-complete problems are reducible to NEW in polynomial time
   a. Describe a reduction function $f$ from a known NP-Complete problem to NEW
   b. Show that $f$ runs in polynomial time
   c. Show that a solution exists to the NP-Complete problem iff a solution exists to the NEW problem generated by $f$

Given a graph $G$, does the graph contain a clique containing exactly half the vertices?

HALF-CLIQUE

1. Show that HALF-CLIQUE is in NP
   a. Provide a verifier
   b. Show that the verifier runs in polynomial time

Verifier: A solution consists of the set of vertices in $V'$
   a. check that $|V'| = |V|/2$
   b. for all pairs of $u, v \in V'$
     * there exists an edge $(u, v) \in E$

   - Check for edge existence in $O(V)$ (assuming adjacency list)
   - $O(V^2)$ checks
   - $O(V^2)$ overall, which is polynomial
Show that all NP-complete problems are reducible to HALF-CLIQUE in polynomial time.

Describe a reduction function \( f \) from a known NP-Complete problem to HALF-CLIQUE.

Show that \( f \) runs in polynomial time.

Show that a solution exists to the NP-Complete problem IFF a solution exists to the HALF-CLIQUE problem generated by \( f \).

Reduce CLIQUE to HALF-CLIQUE:
Given a problem instance of CLIQUE, turn it into a problem instance of HALF-CLIQUE.

\[
\begin{array}{c|c|c}
\text{CLIQUE problem} & \text{HALF-CLIQUE problem} \\
\hline
\text{(Does } G \text{ have a clique of size } k) & \text{(Does } G \text{ have a clique of size } k) \\
\text{yes} & \text{yes} \\
\text{no} & \text{no}
\end{array}
\]

Three cases:

1. \( k = \lfloor |V|/2 \rfloor \)
2. \( k < \lfloor |V|/2 \rfloor \)
3. \( k > \lfloor |V|/2 \rfloor \)

Reduce CLIQUE to HALF-CLIQUE:
Given an instance of CLIQUE, turn it into an instance of HALF-CLIQUE.

It's already a half-clique problem.

\[
\begin{align*}
&1. \text{if } |V| = k \text{ return } G \\
&2. \text{else if } k \leq |V|/2 \\
& \quad \text{return } G + (\lfloor |V|/2 \rfloor - k) \text{ nodes which are fully connected } \\
& \quad \text{and are connected to every node in } V \\
&3. \text{else} \\
& \quad \text{return } G + 2|V| - |V|/2 \text{ nodes which have no edges}
\end{align*}
\]

We're looking for a clique that is smaller than half, so add an artificial clique to the graph and connect it up to all vertices.

\[
\begin{align*}
&1. \text{if } |V|/2 \leq k \text{ return } G \\
&2. \text{else} \\
& \quad \text{return } G + (\lfloor |V|/2 \rfloor - k) \text{ nodes which are fully connected } \\
& \quad \text{and are connected to every node in } V \\
&3. \text{else} \\
& \quad \text{return } G + 2|V| - |V|/2 \text{ nodes which have no edges}
\end{align*}
\]

Reduce CLIQUE to HALF-CLIQUE:
Given an instance of CLIQUE, turn it into an instance of HALF-CLIQUE.
HALF-CLIQUE

Reduce CLIQUE to HALF-CLIQUE:
Given an instance of CLIQUE, turn it into an instance of HALF-CLIQUE.

We're looking for a clique that is bigger than half, so add vertices until \( k = |V|/2 \)

\[
\begin{align*}
\text{\texttt{HALF-CLIQUE}}(G, k) &= \text{\texttt{CLIQUE}}(G + V', k - |V'|/2) \\
\text{\texttt{HALF-CLIQUE}}(G, k) &= \text{\texttt{CLIQUE}}(G + V, k - |V|/2)
\end{align*}
\]

Runtime: From the construction we can see that it is polynomial time.

Reduction proof

Show that a solution exists to the NP-Complete problem \( \text{\texttt{PF}} \) if a solution exists to the new problem generated by \( f \):

- Assume we have an NP-Complete problem instance that has a solution, show that the new problem instance generated by \( f \) has a solution.
- Assume we have a problem instance of NEW generated by \( f \) that has a solution, show that we can derive a solution to the NP-Complete problem instance.

\[
\begin{align*}
\text{\texttt{HALF-CLIQUE}}(G, k) &= \text{\texttt{CLIQUE}}(G + V', k - |V'|/2) \\
\text{\texttt{HALF-CLIQUE}}(G, k) &= \text{\texttt{CLIQUE}}(G + V, k - |V|/2)
\end{align*}
\]
Reduction proof

Given a graph \( G \) that has a CLIQUE of size \( k \), show that \( f(G,k) \) has a solution to HALF-CLIQUE

If \( k < \frac{|V|}{2} \):
- we added a clique of \( |V| - 2k \) fully connected nodes
- there are \( |V| + |V| - 2k = 2|V| - 2k \) nodes in \( f(G) \)
- there is a clique in the original graph of size \( k \)
- plus our added clique of \( |V| - 2k \)
- \( k + |V| - 2k = |V| - k \), which is half the size of \( f(G) \)

Reduction proof

Given a graph \( G \) that has a CLIQUE of size \( k \), show that \( f(G,k) \) has a solution to HALF-CLIQUE

If \( k > \frac{|V|}{2} \):
- we added \( 2k - |V| \) unconnected vertices
- \( f(G) \) contains \( |V| + 2k - |V| = 2k \) vertices
- Since the original graph had a clique of size \( k \) vertices,
  the new graph will have a half-clique

Concrete example

In class is slightly different than what you’d write

I’ve provided a concrete example of the Half-Clique proof on the course webpage
Independent-Set

Given a graph $G = (V, E)$ is there a subset $V' \subseteq V$ of vertices of size $|V'| = k$ that are independent, i.e., for any pair of vertices $u, v \in V'$ there exists no edge between any of these vertices.

Does the graph contain an independent set of size $5$?

Independent-Set

Given a graph $G = (V, E)$ is there a subset $V' \subseteq V$ of vertices of size $|V'| = k$ that are independent, i.e., for any pair of vertices $u, v \in V'$ there exists no edge between any of these vertices.

Independent-Set is NP-Complete

CLIQUE revisited

A clique in an undirected graph $G = (V, E)$ is a subset $V' \subseteq V$ of vertices that are fully connected, i.e., every vertex in $V'$ is connected to every other vertex in $V'$.

CLIQUE problem: Does $G$ contain a clique of size $k$?

Is CLIQUE NP-Complete?

Is CLIQUE NP-Complete?

1. Show that CLIQUE is in NP
   - Provide a verifier
   - Show that the verifier runs in polynomial time
2. Show that all NP-complete problems are reducible to CLIQUE in polynomial time
   - Describe a reduction function $f$ from a known NP-Complete problem to CLIQUE
   - Show that $f$ runs in polynomial time
   - Show that a solution exists to the NP-Complete problem if and only if a solution exists to the CLIQUE problem generated by $f$.

Given a graph $G$, does the graph contain a clique of size $k$?
Independent-Set

Given a graph $G = (V, E)$ is there a subset $V' \subseteq V$ of vertices of size $|V'| = k$ that are independent, i.e., for any pair of vertices $u, v \in V'$ there exists no edge between any of these vertices. Is there an independent set of size $k$?

Independent-Set to Clique

Both are selecting vertices

Independent set wants vertices where NONE are connected

Clique wants vertices where ALL are connected

How can we convert a NONE problem to an ALL problem?

Reduction proof

Show that a solution exists to the NP-Complete problem $P$ if a solution exists to the new problem generated by $f$.

1. Assume we have an Independent-Set problem instance that has a solution, show that the Clique problem instance generated by $f$ has a solution.
   
   Yes \hspace{1cm} Yes

2. Assume we have a problem instance of Clique generated by $f$ that has a solution, show that we can define a solution to Independent-Set problem instance.
   
   Yes \hspace{1cm} \hspace{1cm} Yes

$f(G)$

return $G'$
Proof

Given a graph $G$ that has an independent set of size $k$, show that $f(G)$ has a clique of size $k$

- By definition, the independent set has no edges between any vertices
- These will all be edges in $f(G)$ and therefore they will form a clique of size $k$

Proof

Given $f(G)$ that has clique of size $k$, show that $G$ has an independent set of size $k$

- By definition, the clique will have an edge between every vertex
- None of these vertices will therefore be connected in $G$, so we have an independent set

Independent-Set revisited

Given a graph $G = (V, E)$ is there a subset $V' \subseteq V$ of vertices of size $|V'| = k$ that are independent, i.e. for any pair of vertices $u, v \in V'$ there exists no edge between any of these vertices

Is Independent-Set NP-Complete?

Independent-Set revisited

Given a graph $G = (V, E)$ is there a subset $V' \subseteq V$ of vertices of size $|V'| = k$ that are independent, i.e. for any pair of vertices $u, v \in V'$ there exists no edge between any of these vertices

Reduce 3-SAT to Independent-Set
Given a 3-CNF formula, convert it into a graph

For each clause, e.g., (a OR ~b OR c) create a clique containing vertices representing these literals

- To enforce that only one variable and its complement can be set we connect each vertex representing x to each vertex representing its complement ~x

- for the Independent-Set problem to be satisfied it can only select one variable
- to make sure that all clauses are satisfied, we set k = number of clauses
Given a 3-SAT problem with \( k \) clauses and a valid truth assignment, show that \( f(3\text{-SAT}) \) has an independent set of size \( k \).

(Assume you know the solution to the 3-SAT problem and show how to get the solution to the independent set problem)

Since each clause is an OR of variables, at least one of the three must be true for the entire formula to be true. Therefore each 3-clique in the graph will have at least one node that can be selected.

\[
\begin{align*}
(a \lor \neg b \lor c) \land (c \lor b \lor d) \land (\neg a \lor \neg c \lor \neg d)
\end{align*}
\]

What independent set?
Given a graph with an independent set $S$ of $k$ vertices, show there exists a truth assignment satisfying the boolean formula.

- For any variable $x_i$, $S$ cannot contain both $x_i$ and $\neg x_i$ since they are connected by an edge.

- For each vertex in $S$, we assign it a true value and all others false. Since $S$ has only $k$ vertices, it must have one vertex per clause.
More NP-Complete problems

SUBSET-SUM:
- Given a set S of positive integers, is there some subset S' of S whose elements sum to t?

TRAVELING-SALESMAN:
- Given a weighted graph G, does the graph contain a hamiltonian cycle of length k or less?

VERTEX-COVER:
- Given a graph G = (V, E), is there a subset V' ⊆ V such that if (u,v) ∈ E then u ∈ V' or v ∈ V'?

Our known NP-Complete problems

We can reduce any of these problems to a new problem in an NP-completeness proof

- SAT, 3-SAT
- CLIQUE, HALF-CLIQUE
- INDEPENDENT-SET
- HAMILTONIAN-CYCLE
- TRAVELING-SALESMAN
- VERTEX-COVER
- SUBSET-SUM

Search vs. Exists

All the problems we've looked at asked decision questions:
- Is there a hamiltonian cycle?
- Does the graph have a clique of size k?
- Does the graph have an independent set of size k?
- ...

For many of the problems with a k in them, we really want to know what the largest/smallest one is:
- What is the largest clique in the graph?
- What is the shortest path that visits all the vertices exactly once?

Why don't we care?

P vs. NP

The big question:

Someone finds a polynomial time solution to one of the NP-Complete problems

NP-Complete problems are somehow harder and distinct
Solving NP-Complete problems

[https://www.math.uwaterloo.ca/tsp/](https://www.math.uwaterloo.ca/tsp/)
[https://www.math.uwaterloo.ca/tsp/world/]

Handout

61

62

Is Half-Clique NP-Complete?

1. Show that NEW is in NP
   a. Provide a verifier
   b. Show that the verifier runs in polynomial time

2. Show that all NP-complete problems are reducible to NEW in polynomial time
   a. Describe a reduction function \( f \) from a known NP-complete problem to NEW
   b. Show that \( f \) runs in polynomial time
   c. Show that a solution exists to the NP-complete problem IFF a solution exists to the NEW problem generated by \( f \)

Given a graph \( G \), does the graph contain a clique containing exactly half the vertices?