Hamiltonian Cycles and Traveling Salesman

CSCI 570

Jeffrey Miller, Ph.D.
jeffrey.miller@usc.edu

DISCUSSION 13-SUPPLEMENTAL

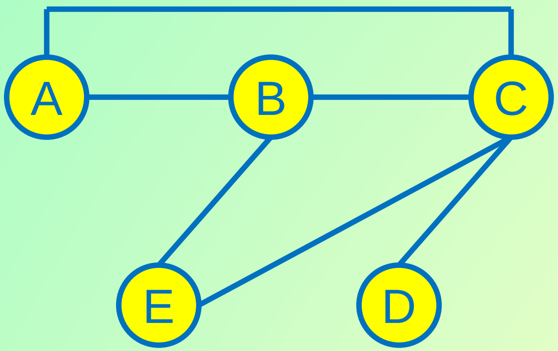
Outline

- Problem Descriptions
  - Hamiltonian Cycle
  - Traveling Salesman Problem
- P, NP, NP-Complete
- Hamiltonian Cycle (HAM-CYCLE)
  - HAM-CYCLE Polynomial-Time Verifiability
  - HAM-CYCLE Reducibility
- Traveling Salesman Problem (TSP)
  - TSP Polynomial-Time Verifiability
  - TSP Reducibility

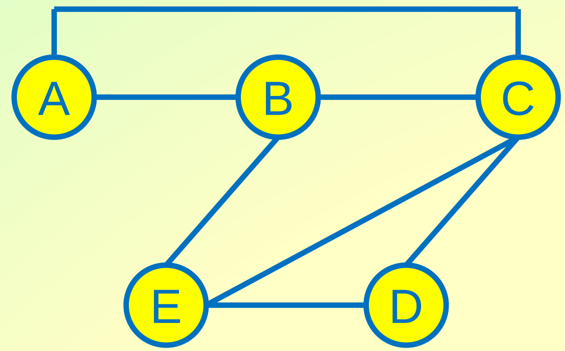
Hamiltonian Cycle Description

- Given an undirected graph G=(V, E), a Hamiltonian Cycle (HAM-CYCLE) is a simple cycle that traverses each vertex $v \in V$
  - A graph is said to be Hamiltonian if it has a Hamiltonian Cycle
  - NOTE: Given an undirected graph G=(V, E), a Hamiltonian Path (HAM-PATH) is a simple path that traverses each vertex $v \in V$
- Optimization Problem
  - An optimization problem seeks to find the **best** solution
  - Given an undirected graph G=(V, E), is the graph Hamiltonian?
- Decision Problem
  - A decision problem answers yes or no
  - Given an undirected graph G=(V, E), is the graph Hamiltonian?
- The optimization and decision problems are the same for HAM-CYCLE because there are no values associated with the problem
  - We have to traverse all the vertices once and only once, other than the start and end vertices
- Formal Description
  - HAM-CYCLE = { <G> : G=(V, E) is an undirected graph,
    G contains a simple cycle that contains all $v \in V$ }

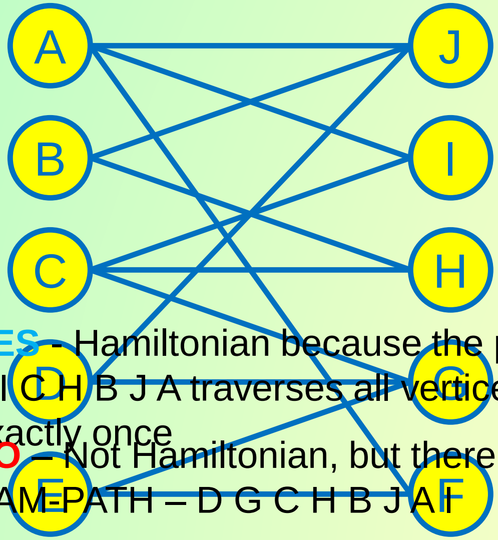
Hamiltonian Cycle Examples



NO - Not Hamiltonian because D only has one edge – no possibility of a simple cycle that traverses all vertices



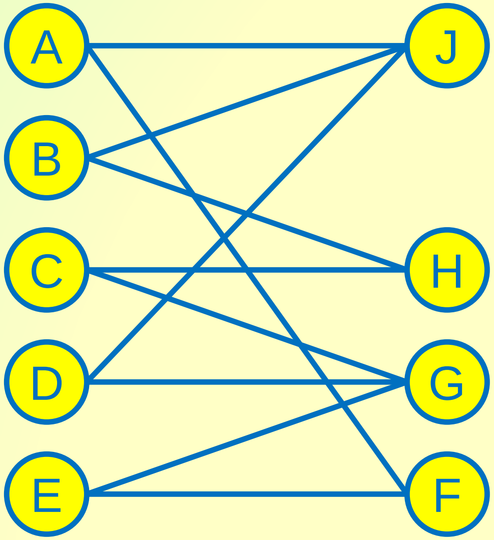
YES - Hamiltonian because the path A B E D C A traverses all vertices exactly once



YES - Hamiltonian because the path A I C H B J A traverses all vertices exactly once

NO – Not Hamiltonian, but there is a HAM-PATH – D G C H B J A I F

YES - Hamiltonian because the path E F A I C H B J D G E traverses all vertices exactly once



NO - Not Hamiltonian because bipartite graphs with an odd number of vertices cannot be Hamiltonian.

Although finding a HAM-CYCLE is NPC, as we will prove today, there are some rules

- Hamiltonian graphs must be biconnected (connected graph with no articulation vertices)
- Bipartite graphs with an odd number of vertices will not be Hamiltonian
- And more…

Traveling Salesman Problem Description

- Given a complete undirected graph G=(V, E) and a weight on each edge, the Traveling Salesman Problem (TSP) finds a Hamiltonian Cycle with minimum overall weight
- Optimization Problem
  - Given a complete undirected graph G=(V, E) with a weight/cost on each edge, what is the Hamiltonian Cycle of minimum weight?
- Decision Problem
  - Given a complete undirected graph G=(V, E) with a weight/cost on each edge, is there a Hamiltonian Cycle of weight less than $k \in \mathbf{Z}$?
- Formal Description
  - TSP = {<G, c, k> : G=(V, E) is a complete undirected graph,
    c is an edge cost function from $V \times V \rightarrow \mathbf{Z}^{+} \cup \{0\}$,
    $k \in \mathbf{Z}$, and
    G has a HAM-CYCLE with cost $\leq k$ }

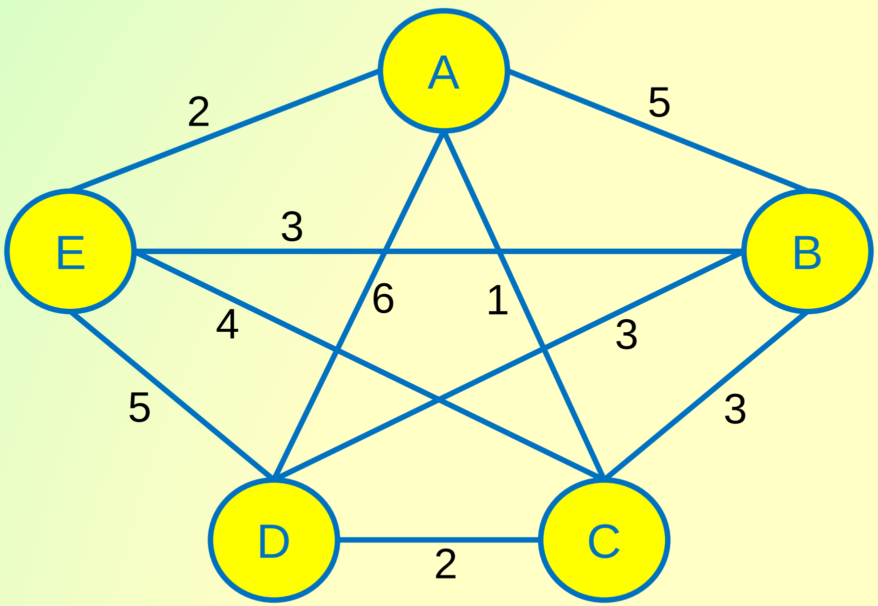
Traveling Salesman Problem Example

A B C D E A = 17
A B C E D A = 23
A B D C E A = 16
A B D E C A = 18
A B E C D A = 20
A B E D C A = 16

A C B D E A = 14
A C B E D A = 18
A C D B E A = 11
A C D E B A = 16
A C E B D A = 17
A C E D B A = 18

A D B C E A = 18
A D B E C A = 17
A D C B E A = 16
A D C E B A = 20
A D E B C A = 18
A D E C B A = 23

A E B C D A = 16
A E B D C A = 11
A E C B D A = 18
A E C D B A = 16
A E D B C A = 14
A E D C B A = 17



We could expand to include all vertices as start and end vertices, but in an undirected graph, cost(ABCDEA) = cost(BCDEAB) = cost(CDEABC) = cost(DEABCD) = cost(EABCDE)

Outline

- Problem Descriptions
  - Hamiltonian Cycle
  - Traveling Salesman Problem
- P, NP, NP-Complete
- Hamiltonian Cycle (HAM-CYCLE)
  - HAM-CYCLE Polynomial-Time Verifiability
  - HAM-CYCLE Reducibility
- Traveling Salesman Problem (TSP)
  - TSP Polynomial-Time Verifiability
  - TSP Reducibility

Polynomial Time (P) Algorithms

- Polynomial Time (P) algorithms are those which have solutions that take polynomial time to run
  - This would be $O(n^k)$ where k is a constant and n is the size of the input
  - How long do Polynomial-time algorithms take to verify?

```
// 0-based array
sort(array A) {
  for (i=1; i < A.length; i++) {
    key = A[i];
    j = i - 1;
    for (j = i-1; j >= 0; j--) {
      if (A[j] <= key) {
        break;
      }
      A[j+1] = A[j];
    }
    A[j+1] = key;
  }
}
```

What sorting algorithm is this?

What is the running time?

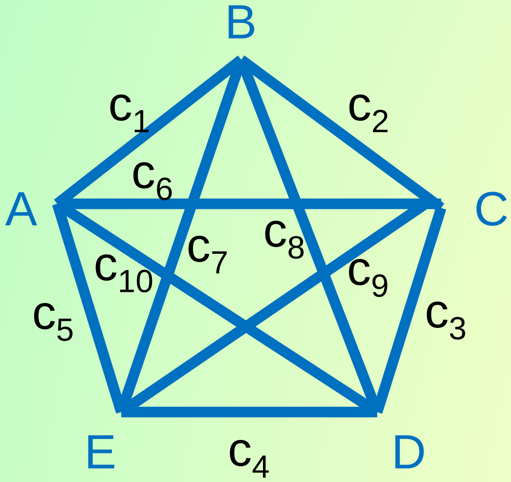
If you were given an array, how long would it take to verify if it was sorted?

```
verify_sorted(array A) {
  for (i=1; i < A.length; i++) {
    if (A[i-1] > A[i]) {
      return "not sorted";
    }
  }
  return "sorted";
}
```

Nondeterministic Polynomial Time (NP) Algorithms

- Nondeterministic Polynomial Time (NP) algorithms are those which have verifiable solutions that take polynomial time to run
  - Given a certificate, the certificate can be verified as a solution in polynomial time
  - Are algorithms in P also in NP?

The problem of finding the minimum path Hamiltonian Cycle (TSP) is in NP. We could enumerate all of the arrangements of vertices that start and end at a specific node and travel through all of the other nodes.



A B C D E A	A C B D E A	A D B C E A	A E B C D A
A B C E D A	A C B E D A	A D B E C A	A E B D C A
A B D C E A	A C D B E A	A D C B E A	A E C B D A
A B D E C A	A C D E B A	A D C E B A	A E C D B A
A B E D C A	A C E B D A	A D E B C A	A E D B C A
A B E C D A	A C E D B A	A D E C B A	A E D C B A

Number of nodes per path = n
Number of paths = (n-1)!
Total running time = O(n!)

Co-Nondeterministic Polynomial Time (Co-NP) Algorithms

- Co-Nondeterministic Polynomial Time (Co-NP) algorithms are the complement algorithms of NP
  - Given a certificate, the certificate can be verified that it is not a solution in polynomial time
  - Are algorithms in co-NP also in NP?
  - Are algorithms in co-NP also in P?

Boolean Satisfiability
Is there a combination of input assignments to satisfy a given boolean formula

Boolean Unsatisfiability
Is there **no** combination of input assignments to satisfy a given boolean formula

z = (a | b) & (~a & b & ~c) & (~a | c) & b

a	b	c	z
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

NP-Complete (NPC) Algorithms

- NP-Complete (NPC) algorithms are NP algorithms that are at least as hard as any other NP-Complete algorithm
  - Polynomial-time verifiable (NP)
  - NP-Complete algorithms can be converted to another NP-Complete algorithm in polynomial time (polynomial-time reducible)
    - A problem is called NP-Hard if it is polynomial-time reducible but not polynomial-time verifiable
- ∴ if any NP-Complete algorithm can be solved in polynomial time, all NP-Complete algorithms can be solved in polynomial time

NP-Complete Problems

CIRCUIT-SAT
Is there a combination of input assignments to satisfy a given boolean circuit

SAT
Is there a combination of input assignments to satisfy a given boolean formula

3-CNF-SAT
Is there a combination of input assignments to satisfy a given 3-CNF boolean formula

CLIQUE
Is there a clique of a given size in a graph

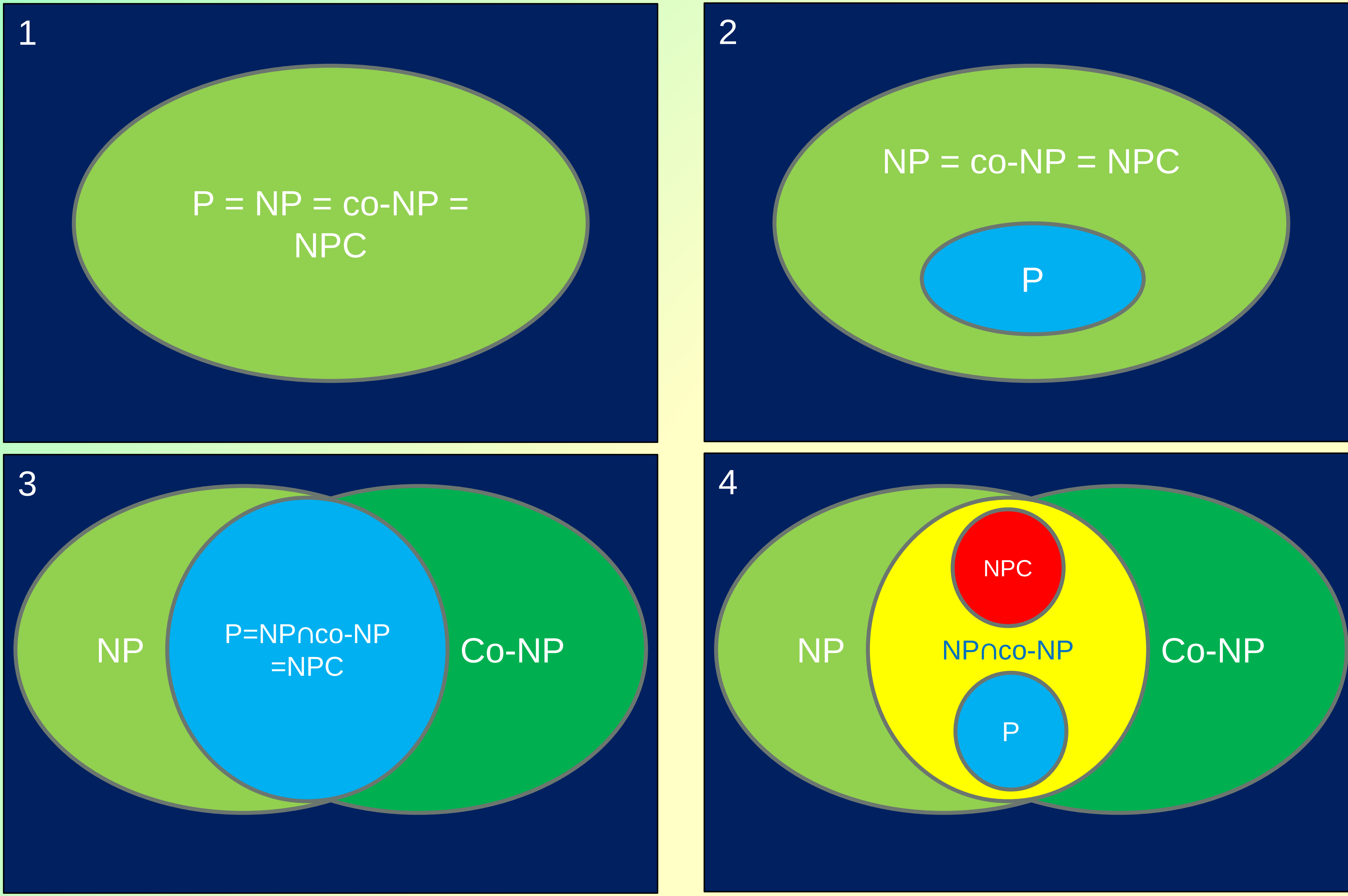
VERTEX-COVER
Is there a set of vertices of a given size that cover all edges in a graph

HAM-CYCLE
Is there a simple cycle that traverses all vertices in a graph

TSP
Is there a Hamiltonian Cycle with a cost not greater than a given value

SUBSET-SUM
Is there a subset of numbers that sum to a specific value given a set of numbers

Relationship of Algorithm Classes



Outline

- Problem Descriptions
  - Hamiltonian Cycle
  - Traveling Salesman Problem
- P, NP, NP-Complete
- Hamiltonian Cycle (HAM-CYCLE)
  - HAM-CYCLE Polynomial-Time Verifiability
  - HAM-CYCLE Reducibility
- Traveling Salesman Problem (TSP)
  - TSP Polynomial-Time Verifiability
  - TSP Reducibility

Hamiltonian Cycle Verifiability

- If we can verify HAM-CYCLE in polynomial time, we will prove that HAM-CYCLE ∈ NP
- Given a graph G=(V, E) where |V|=n and a set of vertices V'={v_0, v_1, v_2,…, v_n}, how do we verify that V' is a Hamiltonian Cycle in G?
  - Number of Vertices
    - Verify that the number of vertices in V' is one more than the number of vertices in G.V and that the start vertex is the same as the end vertex in V' (V_0 = V'_n)
  - Edge Test
    - Verify all of the edges specified in V' exist in G.E
  - All Vertices Only Once
    - Verify all of the vertices in G.V are included in V' with no duplicates (simple cycle)
  - If this all runs in polynomial time, HAM-CYCLE ∈ NP

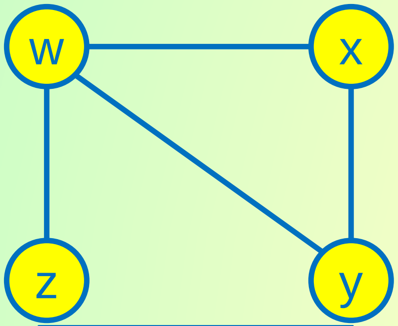
Function	Running Time
Number of Vertices	O(1)
Edge Test	O(n)
All Vertices Only Once	$O(n^2)$
Total	**$O(n^2)$**

Hamiltonian Cycle Verification Code

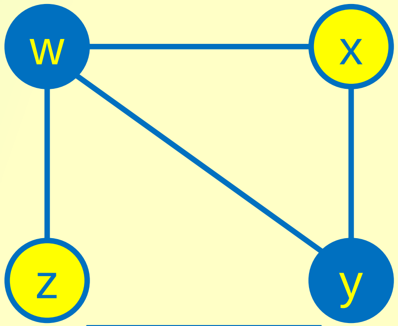
```
# Executions
         boolean verify_HAM_CYCLE(graph G, potential_cycle V’) {
1          if (V’.length != G.V.length + 1 || V’[0] != V’[V’.size-1]) {
1            return false; // if size of V is not same as G.V+1
                           // (start=end), can’t be a HAM-CYCLE
           }
n          for (int i=1; i < V’.length; i++) {
n            if (edge(V’[i-1], V’[i]) ∉ G.E) {
0|1            return false; // an edge in V’ does not exist in G.E
             }
           }
n          for (int j=0; j < G.V.length; j++) {
n            inside = false;
n^2          for (int k=0; k < V’.length-1; k++) {
n^2            if (G.V[j] == V’[k]) {
n                if (inside) {
0|1                return false; // vertex in V’ more than once
                 }
n                inside = true;
               }
             }
n            if (!inside) {
0|1            return false; // a vertex exists in G.V not in V’
             }
           }
0|1        return true; // all vertices traversed with proper edges
         }
```

Vertex Cover Description

- To prove HAM-CYCLE ∈ NP-Complete, we need to reduce another NP-Complete problem to HAM-CYCLE in polynomial time
  - We will use Vertex Cover problem (VERTEX-COVER)
  - Optimization Problem
    - Given an undirected graph G=(V, E), find V* ⊆ V of minimum size such that if (u, v) ∈ E, then u ∈ V* or v ∈ V* or both
  - Decision Problem
    - Given an undirected graph G=(V, E) and k ∈ **Z**, find V* ⊆ V where |V*| ≤ k such that if (u, v) ∈ E, then u ∈ V* or v ∈ V* or both



V={w, x, y, z}

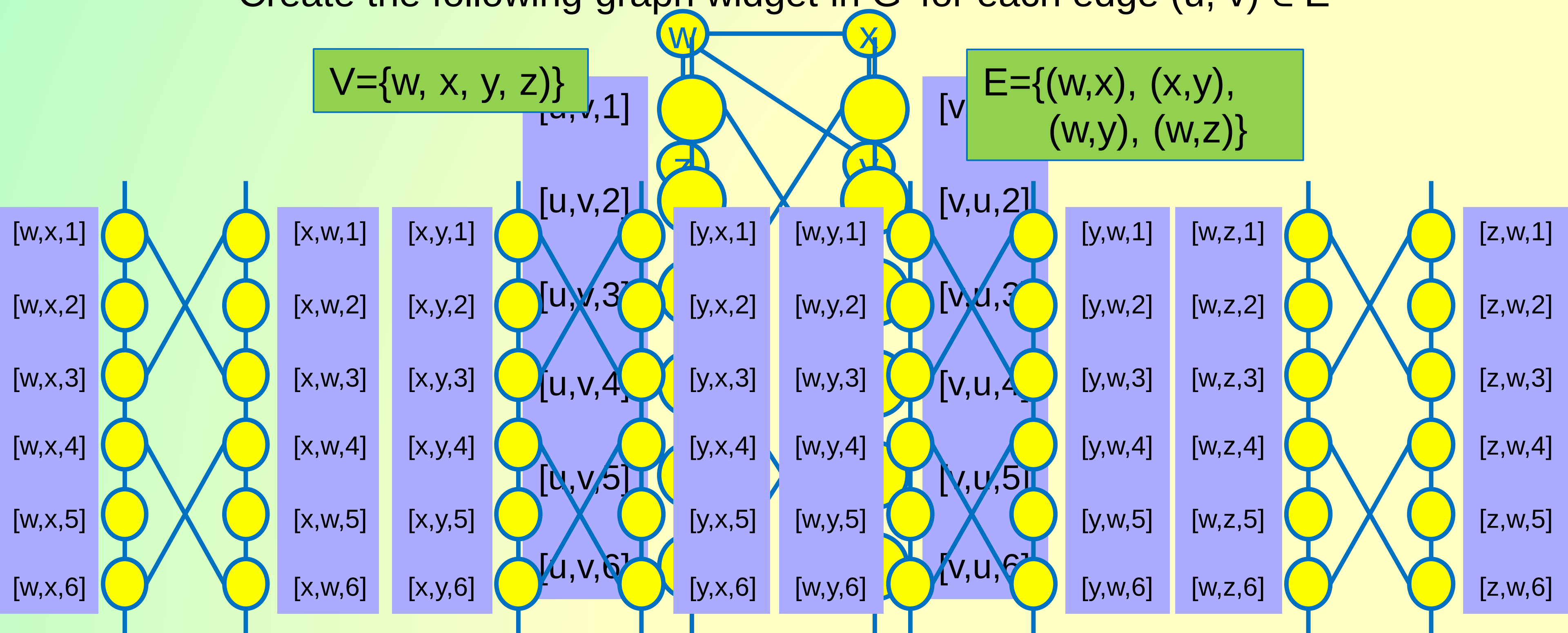


V*={w, y}

w covers (w, x), (w, y), (w, z)
y covers (y, w), (y, x)
NOTE: Both w and y cover (w, y)

Hamiltonian Cycle Reduction

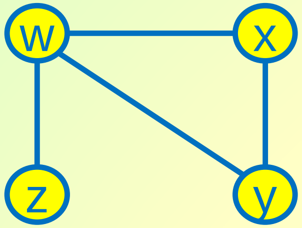
- We need to prove that VERTEX-COVER $\leq_p$ HAM-CYCLE
  - Given an undirected graph G=(V, E) and k∈**Z**, we need to construct an undirected graph G'=(V', E') that has a Hamiltonian Cycle iff G has a vertex cover of size k
  - Step 1
    - Create the following graph widget in G' for each edge (u, v) ∈ E



Hamiltonian Cycle Reduction (cont.)

› Step 2
- Make a list of all vertices adjacent to each vertex in G
  – ∀ u∈V, arbitrarily order adjacent vertices as $u^{(1)}, u^{(2)}, \ldots, u^{(deg(u))}$

V={w, x, y, z)}



E={(w,x), (x,y), (w,y), (w,z)}

u	$u^{(1)}$	$u^{(2)}$	$u^{(3)}$
w	x	y	z
x	w	y	
y	w	x	
z	w		

Note: This example comes from the CLRS textbook. The authors reversed the order of the nodes adjacent to y. This affects future step, so these slides will be slightly different than the book.

u	$u^{(1)}$	$u^{(2)}$	$u^{(3)}$
w	x	y	z
x	w	y	
y	**x**	**w**	
z	w		

Hamiltonian Cycle Reduction (cont.)

› Step 3

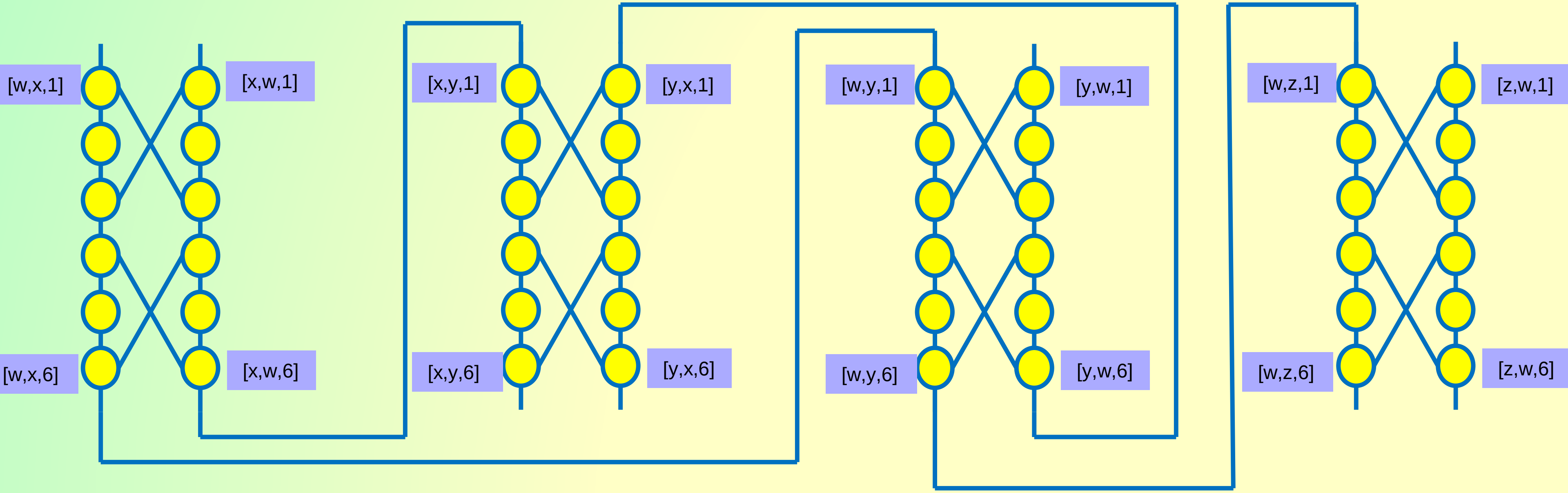
- Connect the widgets in G' based on the following

$\forall$ u$\in$V, add to G' the edges $\{([u,u^{(i)},6], [u,u^{(i+1)},1]) : 1 \leq i \leq \deg(u)\}$

u	$u^{(1)}$	$u^{(2)}$	$u^{(3)}$
w	x	y	z
x	w	y	
y	w	x	
z	w		

Create edges ([w,x,6], [w,y,1]), ([w,y,6],[w,z,1])
Create edge ([x,w,6], [x,y,1])
Create edge ([y,w,6], [y,x,1])

Note: In CLRS, the edge created for y was ([y,x,6], [y,w,1]). The result will be the same, just with different connections.

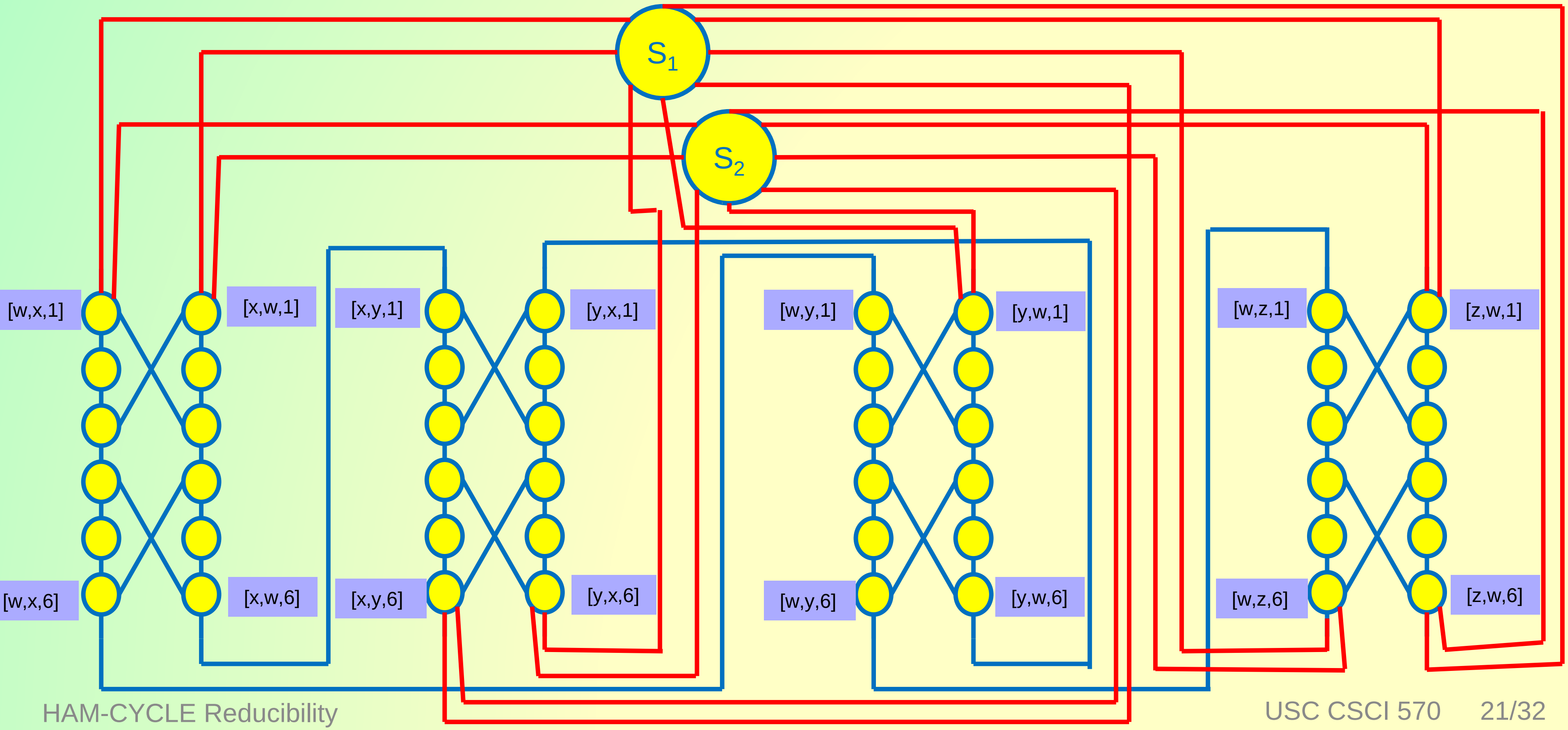


Hamiltonian Cycle Reduction (cont.)

› Step 4

- Create a selector node S_i for each node in the vertex cover set V’
- Connect each of the remaining corners of the widgets that are not already connected to other widgets to all of the S_i selector nodes
  - More formally, add edges to G’ such that

    $\{(s_j,[u,u^{(1)},1]) : u \in V \text{ and } 1 \le j \le k\} \cup \{(s_j,[u,u^{(\deg(u))},6]) : u \in V \text{ and } 1 \le j \le k\}$



Hamiltonian Cycle Reduction (cont.)

V*={w, y}

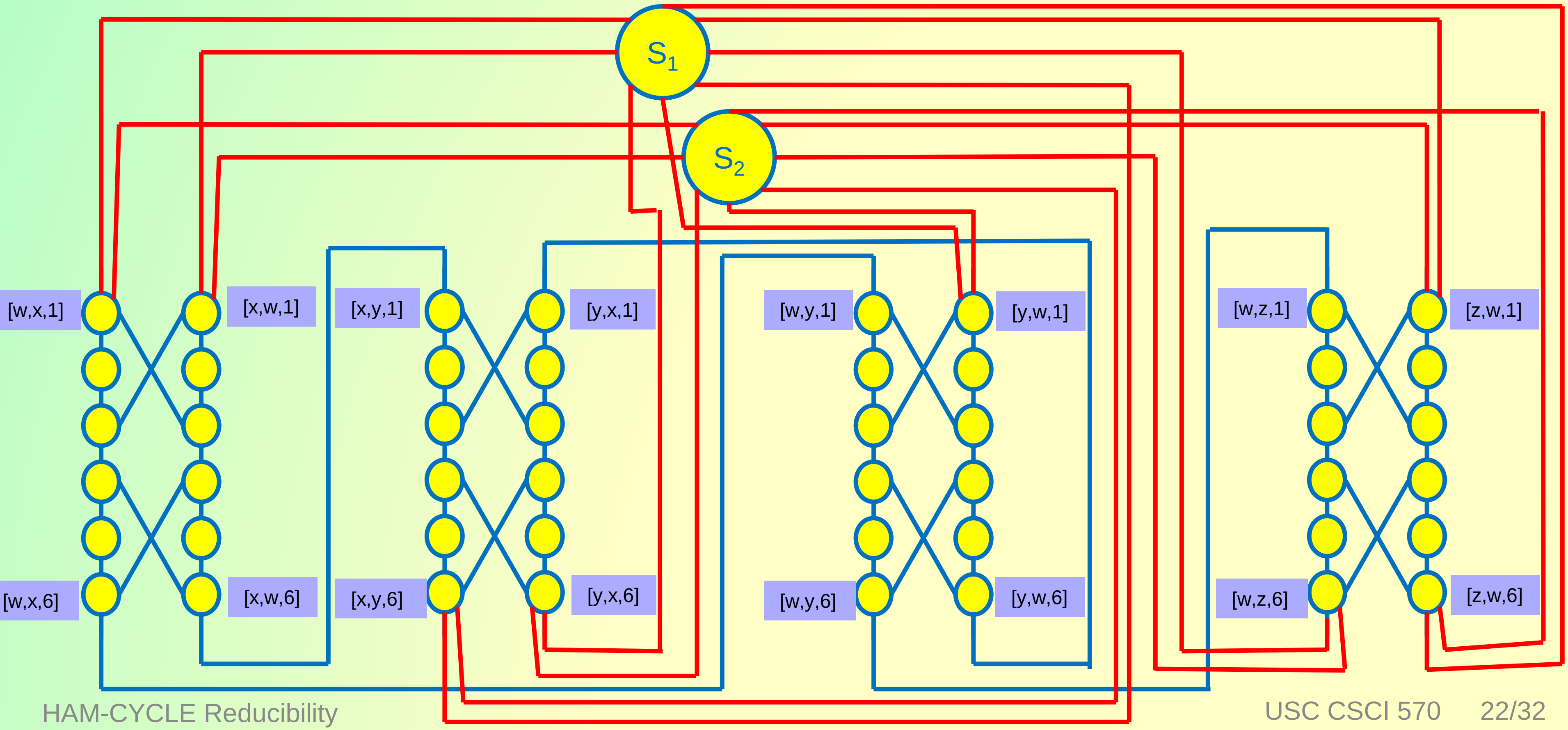
u	$u^{(1)}$	$u^{(2)}$	$u^{(3)}$
w	x	y	z
x	w	y	
y	w	x	
z	w		

› Step 5.1 – Proof

- Assume G=(V, E) has a vertex cover V*⊆V of size k
- A Hamiltonian Cycle in G' includes

$\forall\ u_j \in V^*, \{([u_j,u_j^{(i)},6],[u_j,u_j^{(i+1)},1]) : 1 \le i \le deg(u_j)\}$

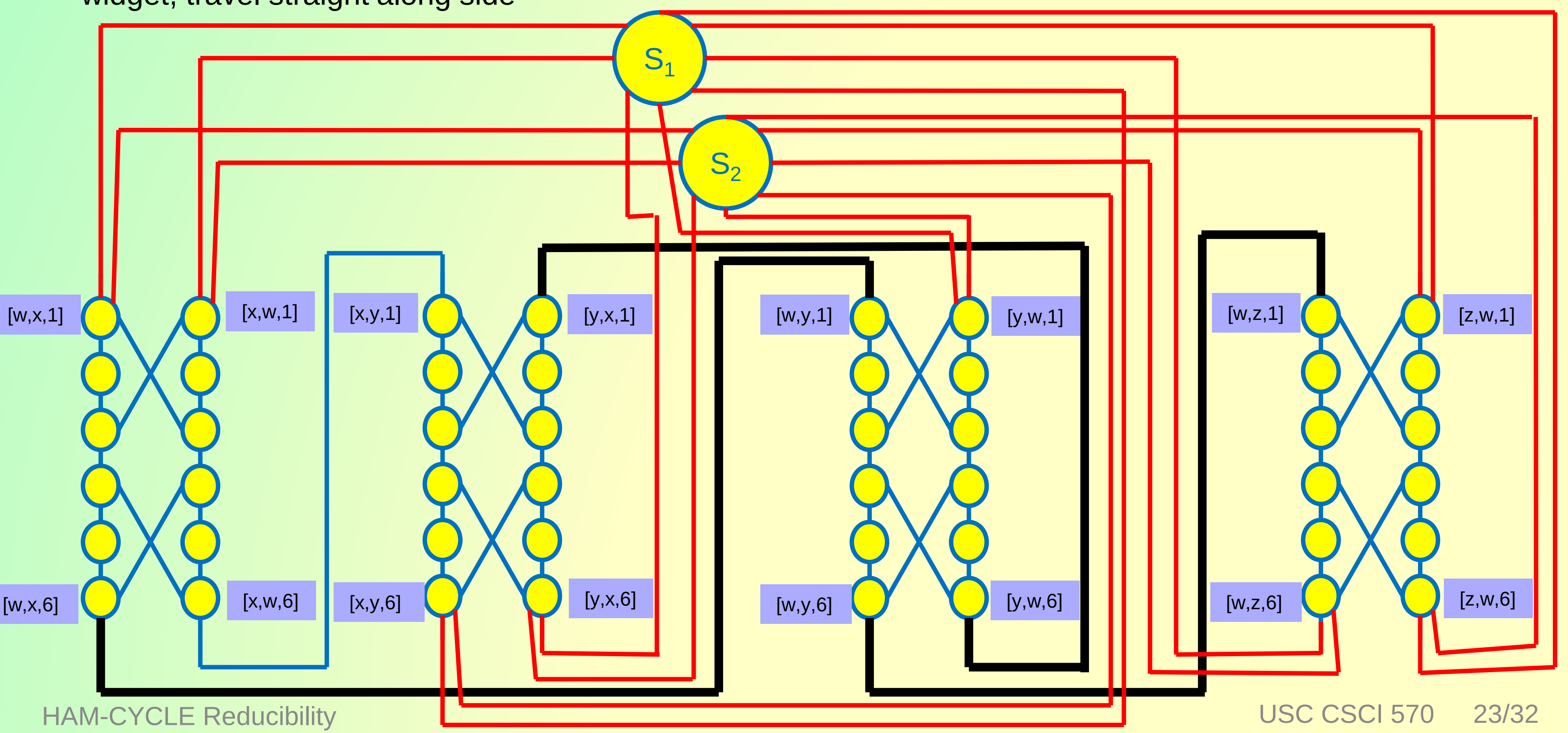
Include ([w,x,6],[w,y,1]), ([w,y,6],[w,z,1])
([y,w,6],[y,x,1])



Hamiltonian Cycle Reduction (cont.)

› Step 5.2 – Proof
  - Assume G=(V, E) has a vertex cover V*⊆V of size k
  - A Hamiltonian Cycle in G' **also** includes edges within each widget that will cover all vertices in the widget from connecting edge already added to G'. If more than one edge is connecting to widget, travel straight along side

Include ([w,x,6],[w,x,1]) covering all nodes
([y,x,1],[y,x,6]) covering all nodes
([w,y,1],[w,y,6]) along side
([y,w,6],[y,w,1]) along side
([w,z,1],[w,z,6] covering all nodes



Hamiltonian Cycle Reduction (cont.)

- Step 5.3 – Proof
  - Assume G=(V, E) has a vertex cover V*⊆V of size k
  - A Hamiltonian Cycle in G' **also** includes

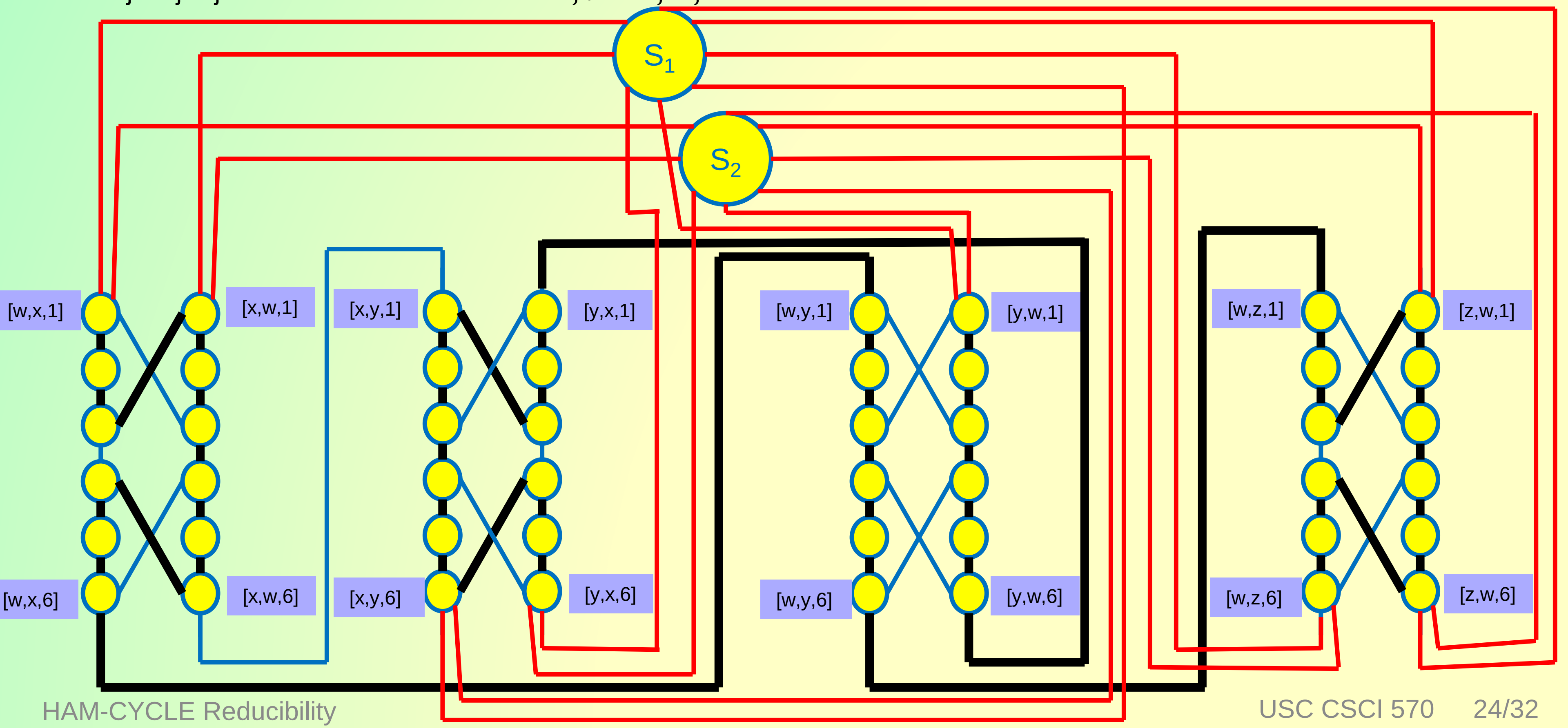
edges from corner nodes of widgets that only have one edge connected to them to specific selector vertices.

V*={w, y}

u	$u^{(1)}$	$u^{(2)}$	$u^{(3)}$
w	x	y	z
x	w	y	
y	w	x	
z	w		

Include (S1,[w,x,1])
(S2,[y,w,1])
(S2, [w,z,6])
(S1,[y,x,6])

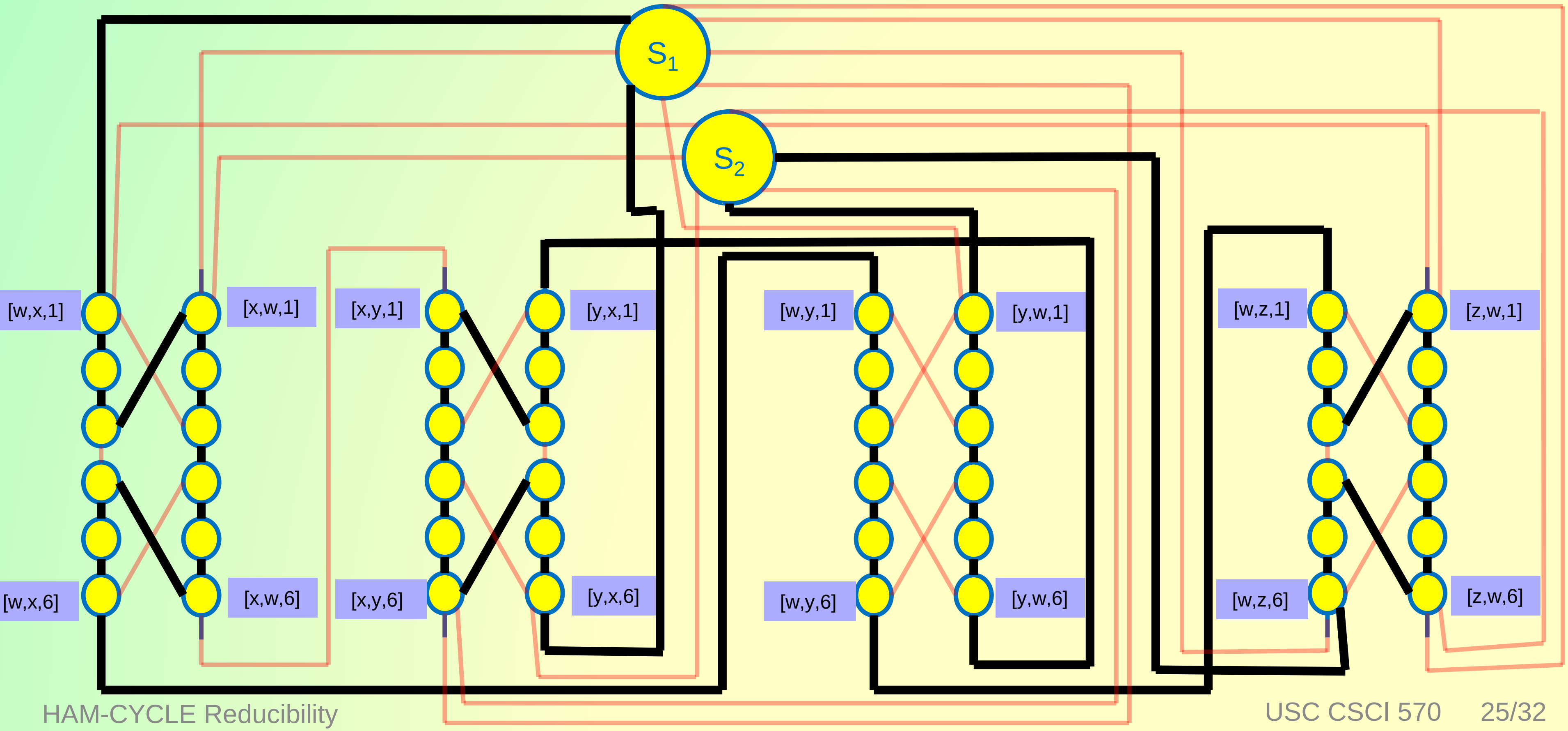
$$\{(s_j,[u_j,u_j^{(1)},1]) : 1 \le j \le k\} \cup \{s_{j+1},[u_j,u_j^{(\deg(uj))},6] : 1 \le j \le k\} \cup \{(s_1,[u_k,u_k^{(\deg(uk))},6])\}$$



Hamiltonian Cycle Reduction (cont.)

› Step 5.4 – Proof

- Assume G=(V, E) has a vertex cover V*⊆V of size k
- As can be seen, the Hamiltonian Cycle starts at S_1, visits all widgets corresponding to edges incident on u_1, visits S_2, visits all widgets with edges incident on u_2, etc., before returning to S_1
- Because V* is a vertex cover for G, each edge in E is incident on some vertex in V*, so the cycle visits each vertex in each widget of G'
- Since the cycle also visits the selector vertices, we have a Hamiltonian Cycle



Hamiltonian Cycle Reduction (cont.)

- Now that we have shown the correctness of the reduction, we just need to show that the reduction can occur in polynomial time
  - To show this, we need to show that the size of G'=(V',E') is polynomial in the size of G=(V,E)
    - Number of vertices in V' is 12 times the numbers of edges, which are how many widgets we have, plus k selector vertices
      - |V'| = 12 |E| + k

        $\boxed{|V'| \le 12\,|E| + |V|}$
    - Edges
      - 14 edges per widget, so we have 14 |E| edges in the widgets
      - For each vertex u∈V, there are deg(u)-1 edges between widgets, which is $\sum_{u \in V}(\deg(u) - 1) = 2|E| - |V|$
      - There are two edges for each pair of selector vertex and each edge in E, which is 2k |E|
      - |E'| = 14|E| + 2|E| - |V| + 2k|E| ≤ 16|E| - |V| + 2|V||E|

        $\boxed{|E'| \le 16|E| + |V|\,(2|E| - 1)}$
  - This shows that the number of vertices |V'| and edges |E'| are polynomial with respect to |V| and |E|

Outline

- Problem Descriptions
  - Hamiltonian Cycle
  - Traveling Salesman Problem
- P, NP, NP-Complete
- Hamiltonian Cycle (HAM-CYCLE)
  - HAM-CYCLE Polynomial-Time Verifiability
  - HAM-CYCLE Reducibility
- Traveling Salesman Problem (TSP)
  - TSP Polynomial-Time Verifiability
  - TSP Reducibility

Traveling Salesman Problem Verifiability

- If we can verify TSP in polynomial time, we will prove that TSP ∈ NP
- Given a graph G=(V, E) where |V|=n and each edge e ∈ E has an associated cost c_e, a set of vertices V'={v_0, v_1, v_2,…, v_n}, and k ∈ **Z**, how do we verify that V' is a solution to the TSP of cost ≤ k in G?
  - Number of Vertices
    - Verify that the number of vertices in V' is one more than the number of vertices in G.V and that the start vertex is the same as the end vertex in V' (V_0 = V'_n)
  - Edge Test
    - Verify all of the edges specified in V' exist in G.E
  - All Vertices Only Once
    - Verify all of the vertices in G.V are included in V' with no duplicates (simple cycle)
  - Edge Cost Test
    - Verify the sum of all the edge costs is less than or equal to k
  - If this all runs in polynomial time, TSP ∈ NP

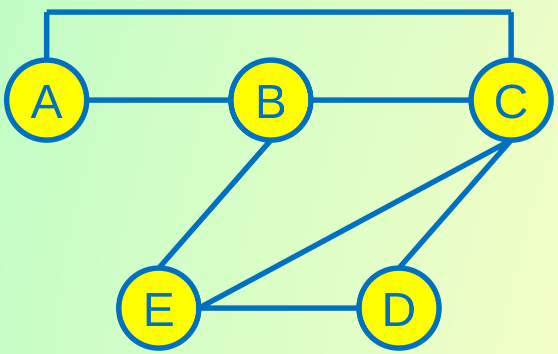
Function	Running Time
Number of Vertices	O(1)
Edge Test	O(n)
All Vertices Only Once	O(n^2)
Edge Cost Test	O(n)
Total	**O(n^2)**

Traveling Salesman Problem Verification Code

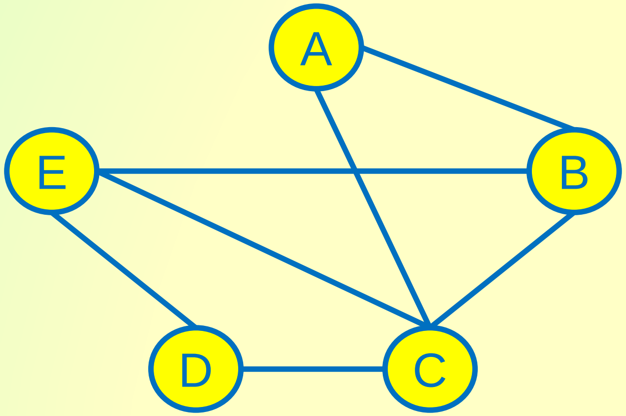
```
# Executions
      boolean verify_TSP(graph G, potential_TSP V', int k) {
1       if (V'.length != G.V.length + 1 || V'[0] != V'[V'.size-1]) {
1         return false; // if size of V is not same as G.V+1
                        // (start=end), can't be a HAM-CYCLE
        }
1       totalCost = 0;
n       for (int i=1; i < V'.length; i++) {
n         if (edge(V'[i-1], V'[i]) ∉ G.E) {
0|1         return false; // an edge in V' does not exist in G.E
          }
n         totalCost += edge(V'[i-1], V'[i]).cost;
        }
1       if (totalCost > k) {
1         return false; // potential TSP has a cost greater than k
        }
n       for (int j=0; j < G.V.length; j++) {
n         inside = false;
n^2       for (int k=0; k < V'.length-1; k++) {
n^2         if (G.V[j] == V'[k]) {
n             if (inside) {
0|1             return false; // vertex in V' more than once
              }
n             inside = true;
            }
          }
n         if (!inside) {
0|1         return false; // a vertex exists in G.V not in V'
          }
        }
0|1     return true; // all vertices traversed with proper edges
      }
```

Traveling Salesman Problem Reducibility

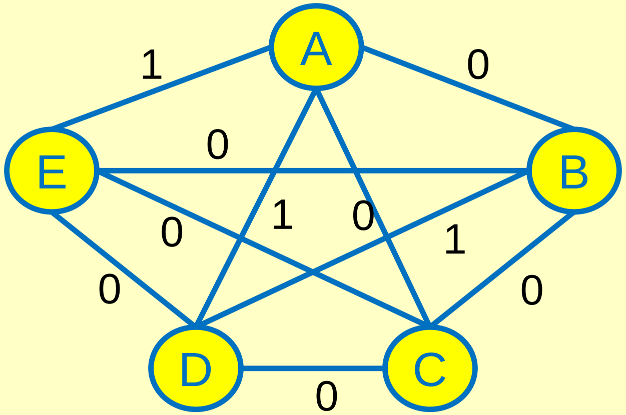
- To prove TSP is NP-Complete, we need to reduce another NP-Complete problem to TSP in polynomial time
  - We will use HAM-CYCLE
  - We need to show that HAM-CYCLE $\leq_p$ TSP
- Let G=(V, E) be an instance of HAM-CYCLE
  - Create G'=(V, E') as a complete graph
    - For each $e \in E'$ where $e \in E$ also, set the cost $c_e = 1$
    - For each $e \in E'$ where $e \notin E$ also, set the cost $c_e = 0$



HAM-CYCLE from earlier slide, G=(V, E)



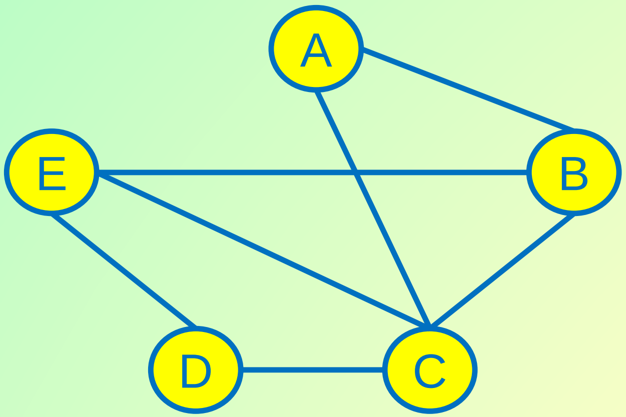
HAM-CYCLE rearranged, G=(V, E)

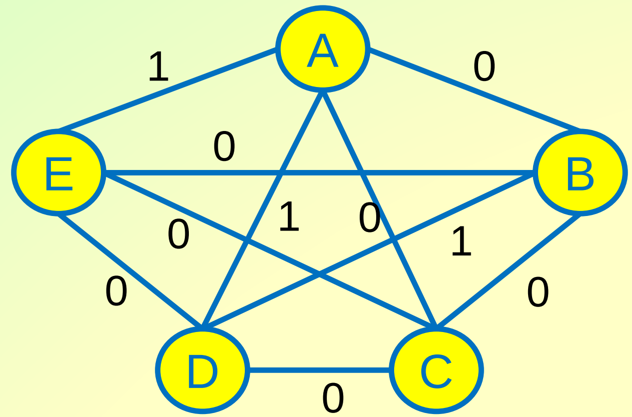


G'=(V, E') based on cost rules above

Traveling Salesman Problem Reducibility (cont.)

G=(V, E)





G'=(V, E')

- Prove that G has a Hamiltonian Cycle iff G' has a Hamiltonian Cycle with cost at most 0 (meaning TSP with cost 0)
  - › Assume G has a Hamiltonian Cycle h
    - Each edge $e \in h$ is also in E ($e \in E$) $\rightarrow$ $c_e=0$ in G'
    - ∴ h is then a Hamiltonian Cycle in G' with cost=0, which is a solution to TSP
  - › Assume G' has a Hamiltonian Cycle h' with cost=0 (a solution to TSP)
    - Since the edge costs in G' are either 0 or 1 and a total cost for h' of 0 $\rightarrow$ all edges in h' have cost 0
    - It follows that h' only contains edges in E since an edge in E' has a cost of 0 only if the edge exists in E
    - ∴ h' is a Hamiltonian Cycle in G

Conclusion

- A Hamiltonian Cycle is a simple cycle in a graph that includes all vertices
- The Traveling Salesman Problem is the Hamiltonian Cycle of minimum cost (where each edge has an associated cost)

Both HAM-CYCLE and TSP are NP-Complete algorithms

- Both algorithms can be **verified in polynomial time**
- Both algorithms can be **reduced in polynomial time** from other NP-Complete algorithms

  VERTEX-COVER $\leq_p$ HAM-CYCLE

  HAM-CYCLE $\leq_p$ TSP