

CS3230 Tutorial 10

Reductions

(AY 25/26 Semester 2)

March 30, 2026

(Prepared by Benson and Hua Jun)

Contents

Recap: Reduction

Tutorial Questions: Reductions

Q1. Graph Coloring

Q2. Partition and Ball Partition

Q3. Partition and Knapsack

Q4. Hamiltonian Cycle and TSP

Bonus. Zero Subset Sum

Recap: Reduction

Main Question: How hard is a problem?



Recap: Reduction

How hard is B.2 in the midterm?

Time to fail this mod




Hard

Should have watched every ted-ed video

leave blank and +0.5

I can't solve it

Very hard


Withdrawal from the University (Undergraduate Students)
Please read the instructions and general notes on this form.

To: Dean of Faculty/Richard

I wish to withdraw from the University for the reason below (please tick only 1 of the following boxes):

<input type="checkbox"/> Unable to cope with studies	<input type="checkbox"/> National Service commitment
<input type="checkbox"/> English language difficulties	<input type="checkbox"/> Obtained employment
<input type="checkbox"/> Financial difficulties	<input type="checkbox"/> Medical reasons
<input type="checkbox"/> Overseas study (self-financing)	<input type="checkbox"/> Not interested in the course
<input type="checkbox"/> Overseas study (scholarship)	<input type="checkbox"/> Personal difficulties
<input type="checkbox"/> Others (please specify): _____	

Name: _____
Student Number: _____
Programme of Study: _____
Current Year of Study: _____
Citizenship: _____
Mailing Address: _____
NUS email: _____
Personal email: _____
Telephone (Home): _____
Telephone (Mobile): _____

Recap: Reduction

What if we don't know whether the problem is solvable?

Time to fail this mod

leave blank and +0.5




I can't solve it

Harder than B.2 in the midterm

Should have watched every ted-ed video

Very hard



Withdrawal from the University (Undergraduate Students)
Please read the instructions and general notes on this form.

To: Dean of Faculty/School

I wish to withdraw from the University for the reason below (please tick only 1 of the following boxes):

<input type="checkbox"/> Unable to cope with studies	<input type="checkbox"/> National Service commitment
<input type="checkbox"/> English language difficulties	<input type="checkbox"/> Obtained employment
<input type="checkbox"/> Financial difficulties	<input type="checkbox"/> Medical reasons
<input type="checkbox"/> Overseas study (self-financing)	<input type="checkbox"/> Not interested in the course
<input type="checkbox"/> Overseas study (scholarship)	<input type="checkbox"/> Personal difficulties
<input type="checkbox"/> Others (please specify): _____	

Name: _____

Student Number: _____

Programme of Study: _____

Current Year of Study: _____

Citizenship: _____

Postal Address: _____

NUS email: _____

Personal email: _____

Telephone (Home): _____

Telephone (Mobile): _____

Recap: Reduction

Consider the following problems:

Problem 1:
 $1 + 1$

Problem 2:

$$\sum_{k=0}^{\infty} k \int_0^{\infty} \frac{(at\lambda)^k}{k!} \mu e^{-(a\lambda+\mu)t} dt$$

Which problem is harder?

- ▶ Those that can solve Problem 2 can solve Problem 1.
- ▶ Most people can solve Problem 1 but not Problem 2.
- ▶ In this situation it is obvious.

Recap: Reduction

Consider the following problems:

Problem 1:
Create a website

Problem 2:
Hire a frontend intern

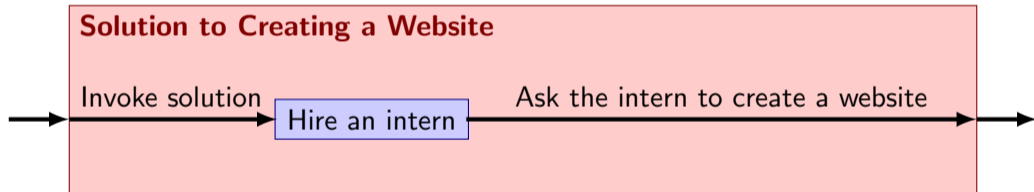
Which problem is harder?

- ▶ If you can hire a frontend intern, you can definitely ask the intern to create a website (if the intern is capable...).
- ▶ Therefore, creating a website is “simpler”.

Recap: Reduction

Main Idea: “We can solve A if we can solve B .”

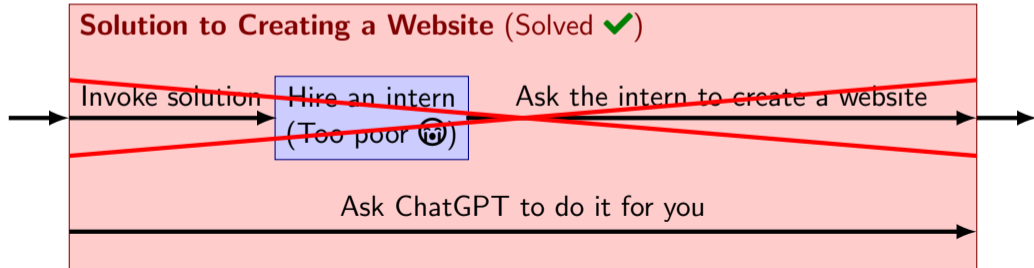
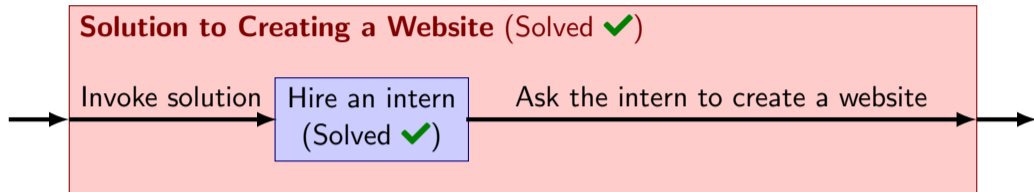
The **reduction** concept:



⚠ “Hire an intern” must be invoked in **black box**. You can’t insert actions during the hiring process, e.g. “Ask the candidates to create a website during their interview”.

Recap: Reduction

What does this imply?



Recap: Reduction

Main Idea: “We can solve A if we can solve B .”

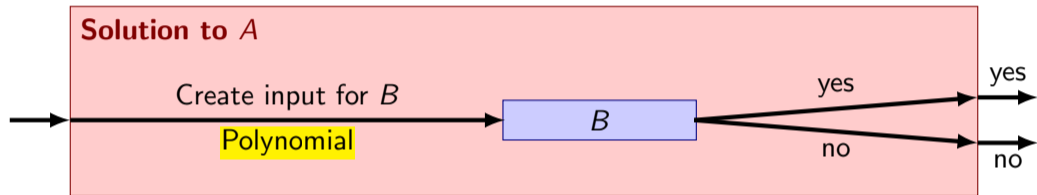
Poll: This implies...

- A. A is easier than B .
- B. A is harder than B .
- C. A is not easier than B .
- D. A is not harder than B .

Solution. D. All solutions that can solve B can be used to solve A as well. Therefore, there are at least as many solutions to A when compared to B .
If A is harder, then there should be some solutions that can solve B but not A .

Recap: Reduction

Definition: We say $A \leq_p B$ when there is a **polynomial-time** reduction from A to B .



Implication:

- ▶ If B can be solved in polynomial time, so do A .
- ▶ If B cannot be solved in polynomial time, we *might* be able to solve A in polynomial time (using other methods).

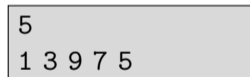
Recap: Reduction

What counts as “polynomial”?

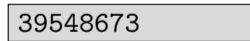
Idea: The runtime of the algorithm must be polynomial to the **size** of the input (not its **magnitude**).

Examples:

- ▶ Finding the maximum of n **digits** in $\Theta(n^2)$
⇒ A polynomial-time algorithm.
 - ▶ The input size is $\Theta(n)$.
- ▶ Checking if p is prime in $\Theta(p)$
⇒ Not a polynomial-time algorithm.
 - ▶ The input size is just $\Theta(\log p)$. An $\Theta(p)$ algorithm is essentially $\Theta(2^{\text{input size}})$.
 - ▶ Note: There is a primality test running in $\tilde{O}(\log^6 p)$.



5
1 3 9 7 5



39548673

size = $\log_{10} p = \Theta(\log p)$

Recap: Reduction

To prove $X \leq_p Y$:

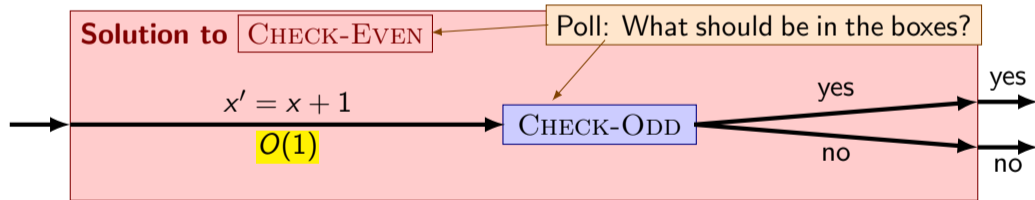
1. Give an algorithm to solve X invoking Y in black-box.
2. Show that the transformation takes polynomial time.
3. Show that a YES-instance for $X \Rightarrow$ YES-instance for Y .
4. Show that a YES-instance for $Y \Rightarrow$ YES-instance for X .

Recap: Reduction

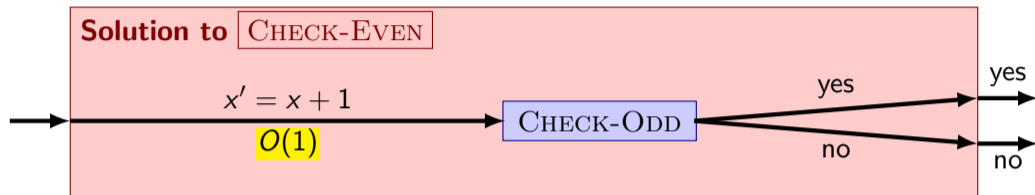
Toy example: Consider the following two problems.

- ▶ CHECK-EVEN: Given an integer x , check whether x is even.
- ▶ CHECK-ODD: Given an integer x , check whether x is odd.

Prove that $\text{CHECK-EVEN} \leq_p \text{CHECK-ODD}$.



Recap: Reduction



To show the correctness: Prove that x is a YES-instance for CHECK-EVEN (i.e. returns true) **if and only if** x' is a YES-instance for CHECK-ODD.

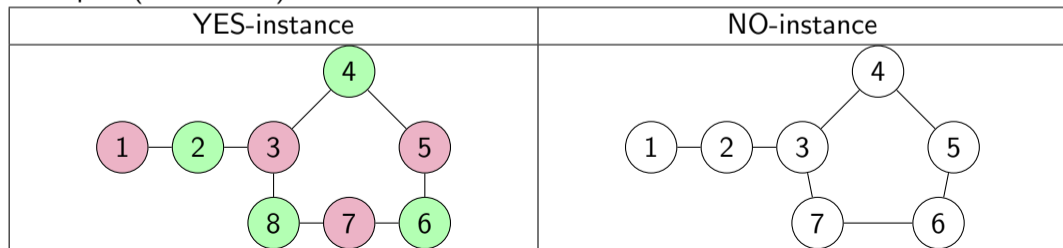
- ▶ (\Rightarrow) Suppose x is a YES-instance for CHECK-EVEN. Then x is even, which means $x' = x + 1$ is odd. Therefore, x' is a YES-instance for CHECK-ODD.
- ▶ (\Leftarrow) Suppose x' is a YES-instance for CHECK-ODD. Then x' is odd, which means $x = x' - 1$ is even. Therefore, x is a YES-instance for CHECK-EVEN.

Q1. Graph Coloring

GRAPH-COLORING: You are given an undirected graph $G = (V, E)$ and asked to color the vertex with a set of colors such that the colors of v_1 and v_2 must be different if $(v_1, v_2) \in E$.

- ▶ Optimization Version: Find the minimum number of colors required to color the graph.
- ▶ Decision Version: Is there a coloring of the graph using k or fewer colors?

Examples (with $k = 2$):



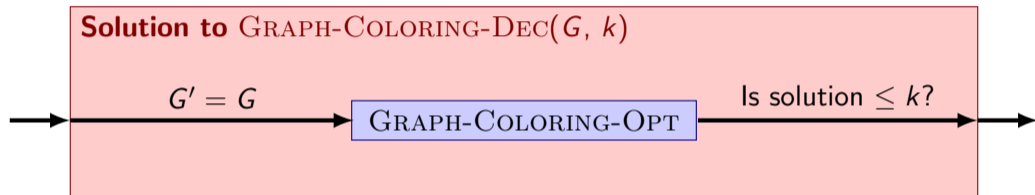
Q1. Graph Coloring

Select all true statements.

- A. If we can solve the optimization problem for graph coloring in polynomial time, we could be able to solve the decision problem for graph coloring in polynomial time.
- B. If we can solve the decision problem for graph coloring in polynomial time, we could be able to solve the optimization problem for graph coloring in polynomial time.
- C. If the decision problem for graph coloring cannot be solved in polynomial time, the optimization problem for graph coloring cannot be solved in polynomial time.
- D. If the optimization problem for graph coloring cannot be solved in polynomial time, the decision problem for graph coloring cannot be solved in polynomial time.

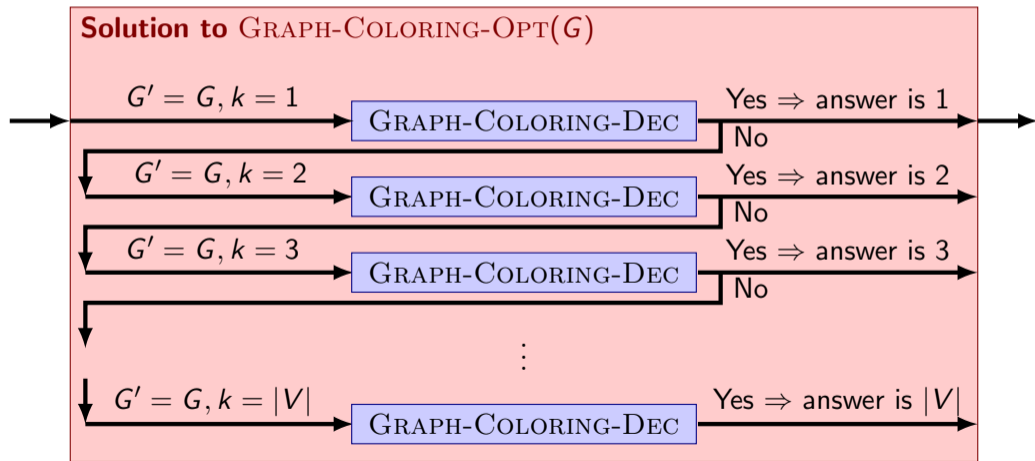
Q1. Graph Coloring

To show that: $\text{GRAPH-COLORING-DEC} \leq_p \text{GRAPH-COLORING-OPT}$.



Q1. Graph Coloring

To show that: $\text{GRAPH-COLORING-OPT} \leq_p \text{GRAPH-COLORING-DEC}$.



Q1. Graph Coloring


Why is this a polynomial-time reduction?

- ▶ We **are** allowed to invoke the blackbox more than once.
- ▶ In this reduction, we are only invoking `GRAPH-COLORING-OPT` $|V|$ times, which is polynomial to the size of the input.
- ▶ Therefore, this is a polynomial-time reduction.
- ▶ **However, this is not a Karp reduction, which is essential in showing NP-completeness (next lecture).**



Q2. Partition and Ball Partition

Consider the following two problems:

- ▶ **PARTITION**: Given a set of **positive** integers S , can this set be partitioned into two sets of equal sum?

YES-instance	NO-instance
	

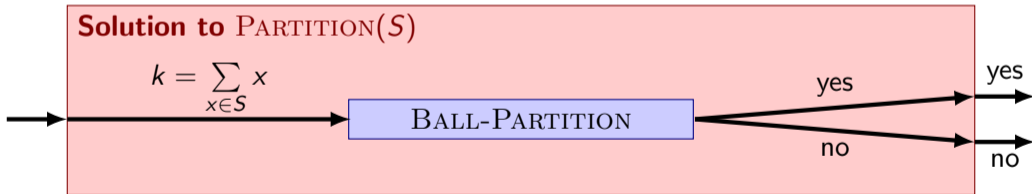
- ▶ **BALL-PARTITION**: Given k balls, can we divide the balls into two boxes with an equal number of balls?

YES-instance	NO-instance
	

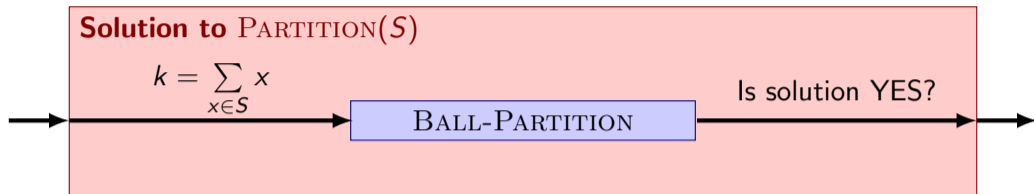
Q2. Partition and Ball Partition

We try to show that $\text{PARTITION} \leq_p \text{BALL-PARTITION}$ using the following transformation A :

1. From the problem PARTITION , we are given S , a set of positive integers.
2. We define k as the total sum of all the elements in S .
3. We use this number k for the BALL-PARTITION problem.



Q2. Partition and Ball Partition



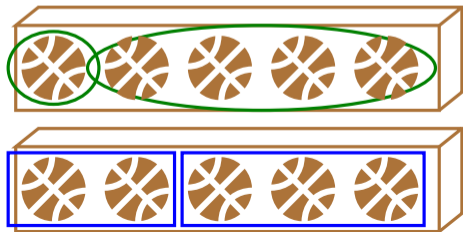
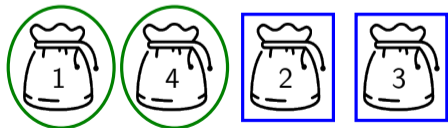
What is wrong with this transformation?

- A. The transformation is correct.
- B. A YES solution to $A(S)$ does not imply a YES solution to S .
- C. A YES solution to S does not imply a YES solution to $A(S)$.
- D. The transformation does not run in polynomial time.

Q2. Partition and Ball Partition

Property 1: A YES solution to $S \Rightarrow$ a YES solution to $A(S)$. ✓ True

- ▶ Given that S is a YES-instance.
- ▶ S can be partitioned into two sets of equal sum, P_1 and P_2 , such that
$$\sum_{x \in P_1} x = \sum_{x \in P_2} x.$$
- ▶ For each $x \in P_1$, take x balls to be placed in the first box. For each $x \in P_2$, take x balls to be placed in the second box.
- ▶ The two boxes have equal number of balls $\Rightarrow A(S)$ is a YES-instance.



Q2. Partition and Ball Partition

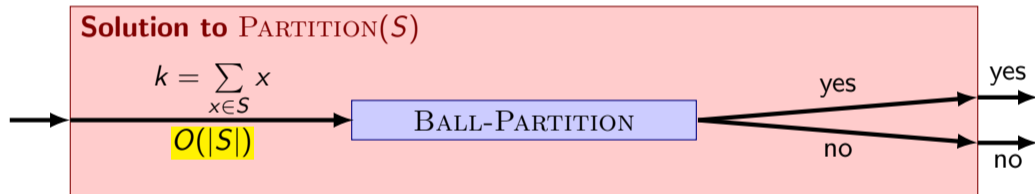
Property 2: A YES solution to $A(S) \Rightarrow$ a YES solution to S . **✗ False**

- ▶ Let $S = \{1, 3\}$.
- ▶ Then $k = A(S) = 4$, which is even $\Rightarrow A(S)$ is a YES-instance.
- ▶ However, S cannot be partitioned $\Rightarrow S$ is a NO-instance.



Q2. Partition and Ball Partition

Property 3: The transformation runs in polynomial time. ✓ True



Q3. Partition and Knapsack

Consider the following two problems:

- ▶ **PARTITION**: Given a set of **non-negative** integers S , can this set be partitioned into two sets of equal sum?
- ▶ **KNAPSACK**: Given a set of n items with **non-negative** weight, value pairs $(w_1, v_1), (w_2, v_2), \dots, (w_n, v_n)$ and a capacity W , is there a subset $I \subseteq \{1, 2, \dots, n\}$ such that $\sum_{i \in I} w_i \leq W$ and $\sum_{i \in I} v_i \geq V$?

Example

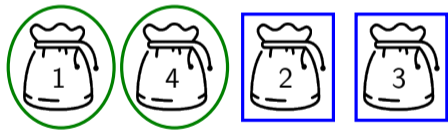


$W = 14$
 $V = 15$

Q3. Partition and Knapsack

We try to show that $\text{PARTITION} \leq_p \text{KNAPSACK}$:

1. Given a PARTITION instance $\{w_1, w_2, \dots, w_n\}$.
2. Define the total sum $S = \sum_{i=1}^n w_i$.
3. Construct a knapsack instance $(w_1, w_1), (w_2, w_2), \dots, (w_n, w_n)$ with capacity $W = S/2$ and threshold $V = S/2$.

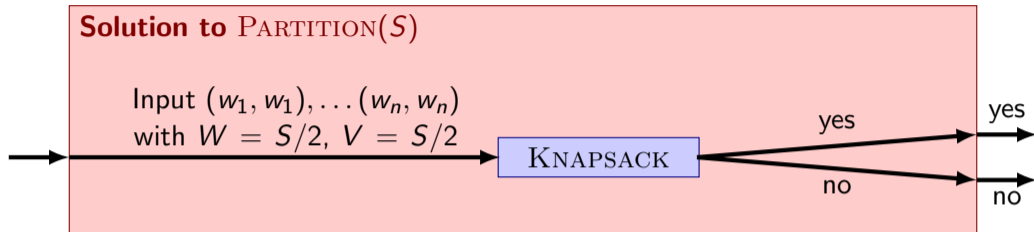


$$S = 1 + 4 + 2 + 3 = 10$$



$$W = 5, V = 5$$

Q3. Partition and Knapsack



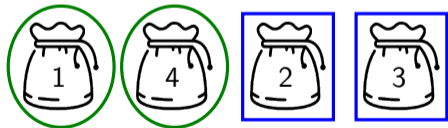
What is wrong with this transformation?

- A. The transformation is correct.
- B. A YES answer to the PARTITION instance does not imply a YES answer to the KNAPSACK instance.
- C. A YES answer to the KNAPSACK instance does not imply a YES answer to the PARTITION instance.
- D. The transformation does not run in polynomial time.

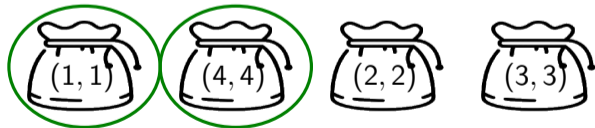
Q3. Partition and Knapsack

Property 1: A YES answer to the PARTITION instance \Rightarrow a YES answer to the KNAPSACK instance. **✓ True**

- ▶ Given that $\{w_1, w_2, \dots, w_n\}$ is a YES-instance, i.e. it can be partitioned into two subsets of sum $S/2$.
- ▶ We take one of the partitions $\{w_{i_1}, w_{i_2}, \dots, w_{i_k}\}$ and select the items $(w_{i_1}, w_{i_1}), (w_{i_2}, w_{i_2}), \dots, (w_{i_k}, w_{i_k})$.
- ▶ Then the sum of weights and values are both $S/2 \Rightarrow$ the KNAPSACK instance is a YES-instance.



$$S = 1 + 4 + 2 + 3 = 10$$

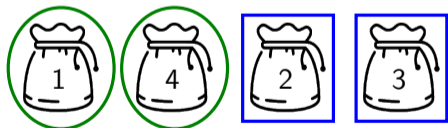


$$W = 5, V = 5$$

Q3. Partition and Knapsack

Property 2: A YES answer to the KNAPSACK instance \Rightarrow a YES answer to the PARTITION instance. **✓ True**

- ▶ Given that the knapsack instance is a YES-instance.
- ▶ Let $(w_{i_1}, w_{i_1}), (w_{i_2}, w_{i_2}), \dots, (w_{i_k}, w_{i_k})$ be the items taken. Since the weight ($\leq S/2$) is equal to the value ($\geq S/2$), we have $w_{i_1} + w_{i_2} + \dots + w_{i_k} = S/2$.
- ▶ Take $\{w_{i_1}, w_{i_2}, \dots, w_{i_k}\}$ as one component and the remaining elements as the other component in the partition. Then both components sum to $S/2 \Rightarrow$ the PARTITION instance is a YES-instance.



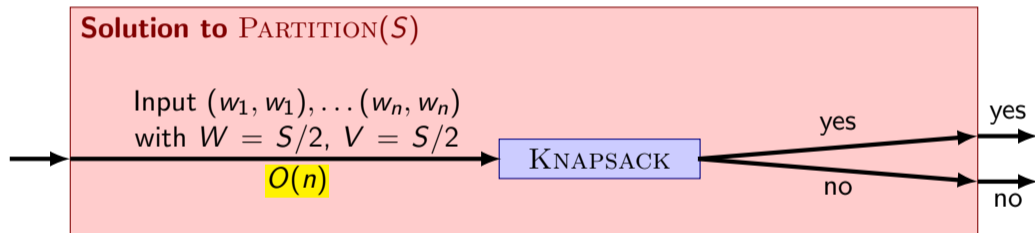
$$S = 1 + 4 + 2 + 3 = 10$$



$$W = 5, V = 5$$

Q3. Partition and Knapsack

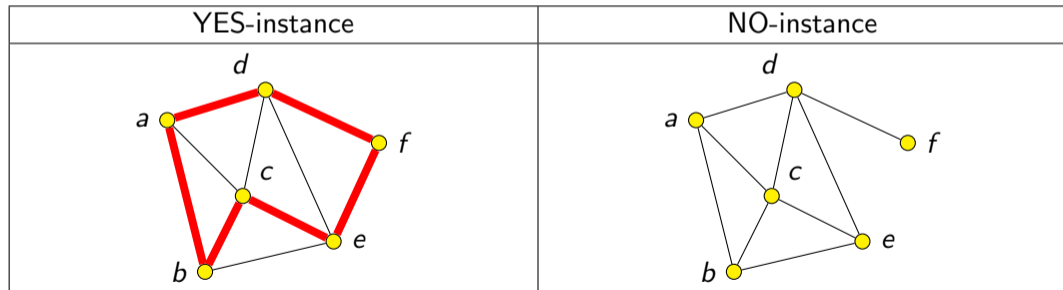
Property 3: The transformation runs in polynomial time. ✓ True



Q4. Hamiltonian Cycle and TSP

HAMILTONIAN-CYCLE: Given an undirected graph $G = (V, E)$, a cycle is said to be **Hamiltonian** if it passes through each vertex exactly once.

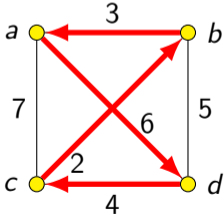
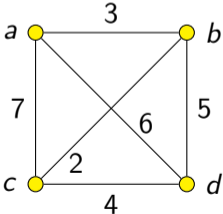
- ▶ Optimization Version: Compute the largest Hamiltonian cycle in G .
- ▶ Decision Version: Does there exist a Hamiltonian cycle of size $|V|$?



Q4. Hamiltonian Cycle and TSP

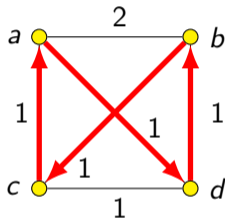
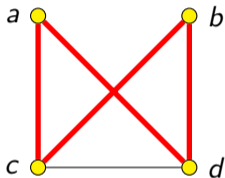
TRAVELLING-SALES-PERSON: Given a weighted, undirected, complete graph $G = (V, E)$ with non-negative cost for its edges. The cost of a tour is the sum of the edge costs traversed in the tour.

- ▶ Optimization Version: Compute the minimum cost tour.
- ▶ Decision Version: Does there exist a tour of cost $\leq b$?

YES-instance	NO-instance
<p data-bbox="136 531 769 572">Here shows a tour with cost ≤ 15.</p> 	<p data-bbox="933 531 1570 572">There are no tours with cost ≤ 14.</p> 

Q4. Hamiltonian Cycle and TSP

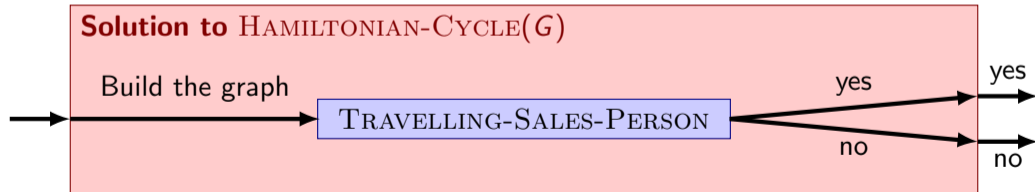
Show that $\text{HAMILTONIAN-CYCLE} \leq_p \text{TRAVELLING-SALES-PERSON}$.



Q4. Hamiltonian Cycle and TSP

Transformation:

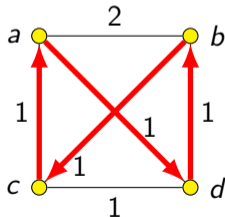
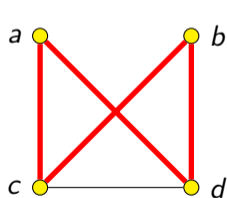
- ▶ Let $G = (V, E)$ be an instance of HAMILTONIAN-CYCLE.
- ▶ Build a complete graph with $|V|$ vertices:
For each pair (u, v) , $\text{cost}(u, v) = \begin{cases} 1 & \text{if } (u, v) \in E \\ 2 & \text{if } (u, v) \notin E \end{cases}$.
- ▶ Set the threshold cost = $|V|$.



Q4. Hamiltonian Cycle and TSP

Property 1: A YES answer to the HAMILTONIAN-CYCLE instance \Rightarrow a YES answer to the TRAVELLING-SALES-PERSON instance. ✔ True

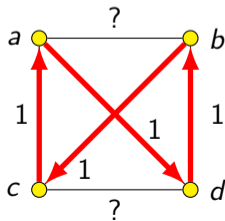
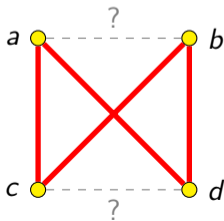
- ▶ Given a Hamiltonian cycle exists in G .
- ▶ The Hamiltonian cycle is a valid TSP tour by definition.
- ▶ Since we set each $\text{cost}(u, v) = 1$ if $(u, v) \in E$, the TSP tour has a cost of $|V|$.
- ▶ \Rightarrow The TRAVELLING-SALES-PERSON instance is a YES-instance.



Q4. Hamiltonian Cycle and TSP

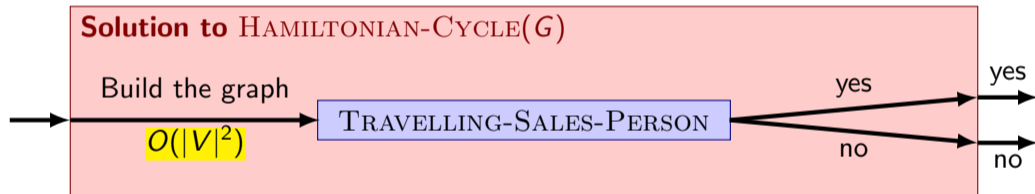
Property 2: A YES answer to the TRAVELLING-SALES-PERSON instance \Rightarrow a YES answer to the HAMILTONIAN-CYCLE instance. **✓ True**

- ▶ Given there exists a tour with cost $|V|$ in the complete graph.
- ▶ Since there are exactly $|V|$ edges in the tour and $\text{cost}(u, v) \geq 1$ by definition, every edge (u, v) in the tour must have $\text{cost}(u, v) = 1$.
- ▶ Thus, $(u, v) \in E$ for all edges (u, v) .
- ▶ The tour is also a valid Hamiltonian cycle. \Rightarrow The HAMILTONIAN-CYCLE instance is a YES-instance.



Q4. Hamiltonian Cycle and TSP

Property 3: The transformation runs in polynomial time. ✓ True



Our progress so far

Every Problem from NP

Lecture 10

CIRCUIT-SAT

CNF-SAT

VERTEX-COVER

INDEPENDENT-SET

3-SAT

SUBSET-SUM

PARTITION

SET COVER

MAX-CLIQUE

HAMILTONIAN-CYCLE

KNAPSACK

FIND-FAMILY

TRAVELLING-SALES-PERSON

An arrow $A \rightarrow B$ indicates that we have proved $A \leq_p B$.

Looking Forward...

Poll: It is already known that 3-SAT is a hard problem. You wish to show that INDEPENDENT-SET is also hard. What should you prove?

- A. $3\text{-SAT} \leq_p \text{INDEPENDENT-SET}$
- B. $\text{INDEPENDENT-SET} \leq_p 3\text{-SAT}$


Answer. $3\text{-SAT} \leq_p \text{INDEPENDENT-SET}$

Solution. To prove that INDEPENDENT-SET is hard, it doesn't make sense to supply a solution to it. Instead, we should show that it can be used (as a blackbox) to solve other hard problems.

Bonus. Zero Subset Sum

Consider the following two problems:

- ▶ **SUBSET-SUM**: Given a set of n non-negative integers $S = \{w_1, \dots, w_n\}$ and a target W , decide whether there exists a subset $I \subseteq \{1, 2, \dots, n\}$ such that $\sum_{i \in I} w_i = W$. (In other words, decide whether there exists a subset of S with sum W .)

YES-instance	NO-instance
<p data-bbox="227 585 354 626">$W = 0$</p>  <p>The YES-instance shows four bags with values 1, 2, -4, and -3. The bags with values 1 and -3 are circled in green, indicating a subset {1, -3} that sums to 0.</p>	<p data-bbox="973 595 1101 637">$W = 4$</p>  <p>The NO-instance shows four bags with values 1, 2, -4, and -3. No subset of these bags sums to 4.</p>

Bonus. Zero Subset Sum

- **ZERO-SUBSET-SUM**: Given a set of n non-negative integers $S = \{w_1, \dots, w_n\}$, decide whether there exists a subset $I \subseteq \{1, 2, \dots, n\}$ such that $\sum_{i \in I} w_i = 0$. (In other words, decide whether there exists a subset of S with sum 0.)

YES-instance	NO-instance
	

Prove that $\text{SUBSET-SUM} \leq_p \text{ZERO-SUBSET-SUM}$.

(Solutions are not provided but you're welcomed to check your ideas with me.)