

TEACHER AND STUDENT RESOURCE

Graph Algorithms and Optimization



DCS.GT.2

Graph Theory

Seven instructional sections. Thirty multiple-choice and thirty open or performance tasks, original mathematical visuals, generous workspaces, teaching scripts, and worked keys.

Free sample | Contents

DCS.GT.2 | NC Discrete Mathematics for Computer Science mathematics

This sample contains 20 complete selected pages from the 103-page resource, plus this sample contents page. Original printed page numbers are retained. Gaps are intentional.

Included original pages: 1-16, 36-37, 66, 81

Representative activities

M01: sample PDF page(s) 16; original page(s) 15

M02: sample PDF page(s) 17; original page(s) 16

O01: sample PDF page(s) 18; original page(s) 36

O02: sample PDF page(s) 19; original page(s) 37

Selected teacher prompts and keys: original page(s) 66, 81

Using this sample

Read the preparation and access guidance before using a selected activity.
Keep teacher keys and private preparation pages separate from student copies.
Use the original resource contents to see the scope of the complete packet.

Complete paid download

The complete resource includes the full PDF task bank and teaching materials.

ALIGNMENT

Standard and learning targets

Apply graph theory to solve problems.

DCS.GT.2.1 Implement critical path analysis algorithms to determine the minimum project time.

DCS.GT.2.2 Implement the brute force method, the nearest-neighbor algorithm, and the cheapest-link algorithm to find solutions to a Traveling Salesperson Problem.

DCS.GT.2.3 Implement vertex-coloring techniques to solve problems.

DCS.GT.2.4 Implement Kruskal and Prim's algorithms to determine the weight of the minimum spanning tree of a connected graph.

The decimal codes are the official objectives. All are represented in the item bank. These materials support several sessions, not one 60-task lesson.

NAVIGATION

Choose evidence deliberately

Warm-up: M01-M04 and O01-O04

I Do: M05-M08 and O05-O08

We Do: M09-M13 and O09-O13

Collaborative practice: M14-M18 and O14-O18

Independent practice: M19-M23 and O19-O23

Small-group reteach: M24-M27 and O24-O27

Exit ticket: M28-M30 and O28-O30

DCS.GT.2.1: M01, M02, M03, M04, M05, M06, M07, O01, O02, O03, O04, O05, O06, O07

DCS.GT.2.2: M08, M09, M10, M11, M12, M13, M14, M15, O08, O09, O10, O11, O12, O13, O14, O15, O16

DCS.GT.2.3: M16, M17, M18, M19, M20, M21, M22, O17, O18, O19, O20, O21, O22, O23

DCS.GT.2.4: M23, M24, M25, M26, M27, M28, M29, M30, O24, O25, O26, O27, O28, O29, O30

DAILY LEARNING TARGETS

A multi-session teaching path

Session 1 Build and interpret representations

Learning target: Implement critical path analysis algorithms to determine the minimum project time. Implement the brute force method, the nearest-neighbor algorithm, and the cheapest-link algorithm to find solutions to a Traveling Salesperson Problem.

Use selected Warm-up and I Do tasks; compare an initial prediction with the worked model.

Session 2 Apply and explain methods

Learning target: Implement vertex-coloring techniques to solve problems. Implement Kruskal and Prim's algorithms to determine the weight of the minimum spanning tree of a connected graph.

Use We Do and collaborative tasks across the objectives. Require each learner to explain one mathematical choice.

Session 3 Transfer and collect independent evidence

Learning target: Apply the listed objectives with accurate representations, justified methods, and contextual checks.

Choose independent tasks covering each objective. Reteach an evidenced gap, then use a fresh exit task without answer-supplying cues.

PLANNING

Materials, access and pacing

Materials

Pencils, erasers, straightedge, printed tasks, and optional scratch paper. A calculator may support arithmetic after the method is stated. Use graph paper or movable labels when useful without supplying the requested construction.

Prerequisites

Weighted graphs, paths, sums, and graph terminology.

Pacing

Select two to four tasks for a session. Preserve first independent evidence, discuss a model, and revisit with a fresh task. The full banks are reusable choices, not a timed requirement.

Neutral access

Read directions aloud without identifying a method or answer. Enlarge diagrams, allow dictated or typed reasoning, and provide accessible tables. Preserve all mathematical conditions.

Keep evidence interpretable

Record neutral supports separately from answer-supplying cues. Retain first responses. A selected distractor is not by itself a diagnosis.

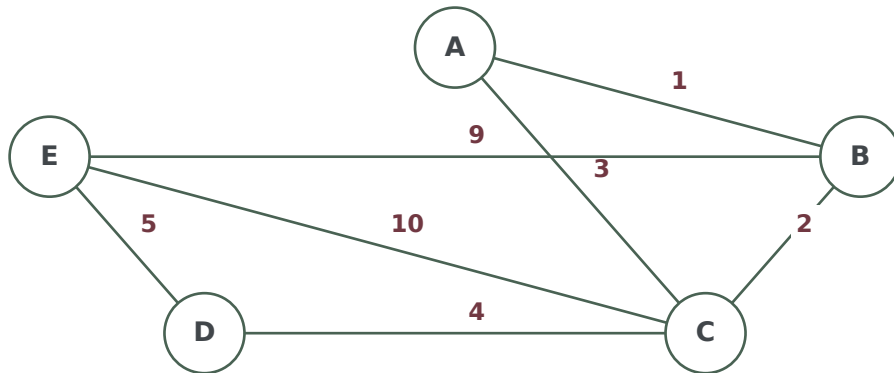
Conventions

A task written A3 means task A has duration 3 time units; an arrow shows a prerequisite. In weighted-edge lists, AB 8 means edge AB has weight 8. Critical paths are longest prerequisite chains with unlimited parallel workers unless stated otherwise. TSP tours are Hamiltonian circuits; a spanning tree has no cycle. Greedy TSP heuristics need not be optimal. Equal-cost choices use alphabetical order when specified.

TEACHER MODEL

Track the greedy rule explicitly

Find an MST of the displayed graph with Kruskal, and verify the result.



Sort by weight: AB 1, BC 2, AC 3, CD 4, DE 5, BE 9, CE 10. Accept AB, BC; reject AC because it closes a triangle; accept CD, DE. The four accepted edges connect all five vertices without a cycle, total 12.

Think aloud: A low weight is considered only after feasibility. I keep components visible, reject a cycle, and stop at a connected $n-1$ -edge tree.

REUSABLE REASONING GUIDE

Represent justify interpret

Givens and conditions

Model and calculation

Justification and independent check

Conclusion in context

WARM-UP

Teaching script

Core 001 | DCS.GT.2.1

Collect a short first response before discussion. Ask learners what is given and which condition controls the method.

Display and ask

For the activity table, compute earliest start and finish of every task, the project duration, and every critical path. Assume unlimited workers and time 0 start.

Work and discuss

A: ES 0, EF 3. B: ES 3, EF 8. C: ES 3, EF 5. D: ES $\max(8,5)=8$, EF 12. Project duration 12; unique critical path A-B-D totals $3+5+4$. A-C-D totals 9 and is not critical.

Listen for evidence

Complete forward pass and path comparison.

Probe and respond

Ask which objective is being minimized and which feasibility restrictions apply. A low numerical total does not validate an illegal tour, coloring, or disconnected network. Compare the actual chosen edges with the rule.

Next: use M01 for contrast after discussion. Select another task from M01-M04 or O01-O04 to check transfer. Record support and the new evidence separately.

I DO

Teaching script

Core O05 | DCS.GT.2.1

Make a prediction before modeling. Think aloud through the core task, making each calculation and check visible.

Display and ask

Using A2,B5,C3 after A,D4 after A&B,E2 after C&D, compare shortening A by 1 with shortening B by 1. Calculate both new project durations and explain the difference.

Work and discuss

Original duration 11. If A becomes 1, C finishes 4,D still waits for B5 and finishes 9,E finishes 11. If B becomes 4, D starts 4, finishes 8; C still finishes 5; E finishes 10. A is not controlling the D join, while B is on the critical path.

Listen for evidence

Recompute both cases rather than assuming all task reductions save project time.

Probe and respond

Ask which objective is being minimized and which feasibility restrictions apply. A low numerical total does not validate an illegal tour, coloring, or disconnected network. Compare the actual chosen edges with the rule.

Next: use M05 for contrast after discussion. Select another task from M05-M08 or O05-O08 to check transfer. Record support and the new evidence separately.

WE DO

Teaching script

Core 009 | DCS.GT.2.2

Ask learners to propose the next step and have a partner verify it against the original conditions.

Display and ask

Run nearest-neighbor on the cost table starting at A. Record the available unvisited choices at each step, the final route, and total. Compare with the brute-force optimum 24.

Work and discuss

At A: choose B8 over D10,C11. At B: choose D1 over C2. From D only C remains, cost 7; return C-A11. Route A-B-D-C-A costs 27, which is 3 above optimum 24. A local cheapest choice does not ensure a global optimum.

Listen for evidence

Full greedy trace, closing edge, and quantified optimality gap.

Probe and respond

Ask which objective is being minimized and which feasibility restrictions apply. A low numerical total does not validate an illegal tour, coloring, or disconnected network. Compare the actual chosen edges with the rule.

Next: use M09 for contrast after discussion. Select another task from M09-M13 or O09-O13 to check transfer. Record support and the new evidence separately.

COLLABORATIVE PRACTICE

Teaching script

Core O14 | DCS.GT.2.2

Assign reader, modeler, checker, and explainer roles. Require a private first attempt before comparing approaches.

Display and ask

Construct a complete four-vertex weighted graph where nearest-neighbor from A is not optimal. Give its costs, the heuristic route, and a cheaper verified tour.

Work and discuss

One example: AB 2, AC 3, AD 4, BC 6, BD 7, CD 21. Nearest-neighbor from A is A-B-C-D-A, costing $2+6+21+4=33$. Tour A-C-B-D-A costs $3+6+7+4=20$. The cheaper tour is enough to disprove optimality of the heuristic result; no triangle inequality is assumed.

Listen for evidence

A complete specified graph, lawful heuristic trace, and a strictly cheaper valid circuit.

Probe and respond

Ask which objective is being minimized and which feasibility restrictions apply. A low numerical total does not validate an illegal tour, coloring, or disconnected network. Compare the actual chosen edges with the rule.

Next: use M14 for contrast after discussion. Select another task from M14-M18 or O14-O18 to check transfer. Record support and the new evidence separately.

INDEPENDENT PRACTICE

Teaching script

Core O19 | DCS.GT.2.3

Provide only neutral access supports. Keep the first response and ask for justification without selecting a method.

Display and ask

A graph has bipartition $\{A,B,C\}$ and $\{D,E\}$, with every cross-part edge and no within-part edges. Find a minimum coloring and explain why the size 3 part does not require 3 colors.

Work and discuss

Assign color 1 to A,B,C and color 2 to D,E. No within-part pair is adjacent, and every edge crosses colors, so 2 suffice. At least 2 are needed because an edge exists. Members of a part may share a color regardless of that part's size.

Listen for evidence

Use adjacency rather than group size to determine color constraints.

Probe and respond

Ask which objective is being minimized and which feasibility restrictions apply. A low numerical total does not validate an illegal tour, coloring, or disconnected network. Compare the actual chosen edges with the rule.

Next: use M19 for contrast after discussion. Select another task from M19-M23 or O19-O23 to check transfer. Record support and the new evidence separately.

SMALL-GROUP RETEACH

Teaching script

Core O24 | DCS.GT.2.4

Use the earlier work to identify a specific gap. Model that connection, then remove the cue on a fresh item.

Display and ask

Run Kruskal's algorithm on the displayed graph. List accepted and rejected edges until a spanning tree is complete, then verify its edge count, connectivity, and total weight.

Work and discuss

AB 1 accept; BC 2 accept; AC 3 reject because A-B-C-A would be a cycle; CD 4 accept; DE 5 accept. Now 5 vertices are connected by 4 edges, total $1+2+4+5=12$. Stop; larger BE 9, CE 10 are unnecessary.

Listen for evidence

Sorted-edge trace, cycle test, and all three tree checks.

Probe and respond

Ask which objective is being minimized and which feasibility restrictions apply. A low numerical total does not validate an illegal tour, coloring, or disconnected network. Compare the actual chosen edges with the rule.

Next: use M24 for contrast after discussion. Select another task from M24-M27 or O24-O27 to check transfer. Record support and the new evidence separately.

EXIT TICKET

Teaching script

Core O28 | DCS.GT.2.4

Collect a concise claim and supporting computation before showing the key. Use the evidence to select a next task.

Display and ask

Create a connected weighted graph with two different minimum spanning trees. Exhibit both trees and prove their common weight is minimum.

Work and discuss

Example triangle ABC with every edge weight 1. Any two edges form a spanning tree; $\{AB, BC\}$ and $\{AB, AC\}$ are different and both total 2. Every spanning tree on 3 vertices has 2 edges, each weight 1, so no lower total is possible.

Listen for evidence

A tied-optimum construction and a genuine lower bound.

Probe and respond

Ask which objective is being minimized and which feasibility restrictions apply. A low numerical total does not validate an illegal tour, coloring, or disconnected network. Compare the actual chosen edges with the rule.

Next: use M28 for contrast after discussion. Select another task from M28-M30 or O28-O30 to check transfer. Record support and the new evidence separately.

MULTIPLE-CHOICE BANK

M01 Select and justify

M01 | Warm-up | DCS.GT.2.1

Use the activity table with unlimited parallel workers. What is the earliest project finish time?

Task	Duration	Predecessors
A	3	none
B	5	A
C	2	A
D	4	B and C

- A. 14
- B. 8
- C. 9
- D. 12

Reasoning: _____

MULTIPLE-CHOICE BANK

M02 Select and justify

M02 | Warm-up | DCS.GT.2.1

In the same activity table, when can D start at the earliest?

Task	Duration	Predecessors
A	3	none
B	5	A
C	2	A
D	4	B and C

- A. 3
- B. 13
- C. 5
- D. 8

Reasoning: _____

WARM-UP

001 Show your reasoning

DCS.GT.2.1 | Implement critical path analysis algorithms to determine the minimum project time.

For the activity table, compute earliest start and finish of every task, the project duration, and every critical path. Assume unlimited workers and time 0 start.

Task	Duration	Predecessors
A	3	none
B	5	A
C	2	A
D	4	B and C

Representation and calculation

Explanation and check

WARM-UP

002 Show your reasoning

DCS.GT.2.1 | Implement critical path analysis algorithms to determine the minimum project time.

For $A3 \rightarrow \text{parallel } B5, C2 \rightarrow D4$, calculate C's latest start and slack if project deadline is the earliest finish. Explain the effect of delaying C by 2 and by 4 units.

Representation and calculation**Explanation and check**

WORKED TEACHER KEYS

M01 / M02**M01 | DCS.GT.2.1****Answer D: 12**

A finishes 3; B finishes 8, C 5; D starts $\max(8,5)=8$ and finishes 12.

- A. This adds parallel branch durations B and C.
 - B. This omits final task D.
 - C. This uses only the shorter A-C-D branch.
 - D. Correct. A-B-D has duration $3+5+4=12$ and controls the join.
- Evidence: Use maximum predecessor finish at a join.

M02 | DCS.GT.2.1**Answer D: 8**

D's earliest start is the later predecessor finish, 8. Parallel predecessors do not have their finish times added.

- A. Only A has finished at 3.
 - B. This adds both predecessor finish times rather than taking their maximum.
 - C. C is done then, but B is not.
 - D. Correct. It waits for both B and C; their finishes are 8 and 5.
- Evidence: Join-time interpretation.

WORKED TEACHER KEYS

001 / 002**001 | DCS.GT.2.1**

A: ES 0, EF 3. B: ES 3, EF 8. C: ES 3, EF 5. D: ES $\max(8,5)=8$, EF 12. Project duration 12; unique critical path A-B-D totals $3+5+4$. A-C-D totals 9 and is not critical.

Evidence: Complete forward pass and path comparison.

002 | DCS.GT.2.1

Deadline 12 makes D's latest start 8. C must finish by 8, so latest start 6. Earliest start 3, giving slack 3. A 2-unit delay starts C 5, finishes 7, so no project delay. A 4-unit delay finishes 9, delays D to 9, and finishes project 13.

Evidence: Backward timing and two tested delays relative to slack.