mytekDev

Telescope Lab: The Night's Queue, a Greedy Scheduler

Grades 9-12 · 45 minutes

Computer science



Slides, rubric and answer key **Print and go** **No student accounts**

Chromebook friendly

What's included

- A one-page overview: objectives, essential question, vocabulary, standards and assessment
- A 45-minute lesson sequence: warm-up, direct instruction, guided practice, independent practice, closure
- Slides for the board: a PowerPoint file that also opens in Google Slides
- A student worksheet with a QR code and the scheduler program and a table for five versions
- A 3-question exit ticket, two to a page, with an answer key
- A rubric: 3 criteria, 4 levels, scored out of 12
- Differentiation for English learners, students with IEPs or 504 plans, and students who need more support or more challenge
- Cross-curricular extensions, and notes for teaching it in middle school and in AP Computer Science
- Standards: CSTA 2017

Telescope Lab: The Night's Queue, a Greedy Scheduler

Grades 9-12 · 45 minutes · Computer science

An observatory has one night and more targets than it can take in any order. In Level 8 of Telescope Lab seven targets are worth 23 priority points, and each counts only if it stays above 30 degrees for the whole hour or so it needs. The starter takes them in the order of the list: five have not risen yet, each failed try costs 10 minutes, and a for loop never comes back to them, so it ends at 9:55 pm with 6 points. Students write a greedy scheduler: each time the telescope is free, observe the highest priority target that is not done and stays high long enough, or wait 10 minutes and look again. It collects all 23. Then they take the rule apart, one piece at a time, and find which pieces did the work tonight.

At a glance

ESSENTIAL QUESTION

How does a program choose well, one step at a time?

OBJECTIVES

Students will be able to:

- describe a greedy algorithm: at each step, the best choice that is possible now
- write a loop that selects a best item from a list by a condition and a comparison
- explain why the list-order program scores 6 and the scheduler 23
- test a program with one part removed at a time and record what changes
- justify the choice of a while loop with a selection inside over a single for loop

VOCABULARY

Greedy algorithm, Priority, Constraint, Altitude, Block, Selection

STANDARDS

3A-AP-13, 3A-AP-15, 3B-AP-10

TIME AND MATERIALS

45 minutes, grades 9-12.

- One computer or Chromebook per student
- A projector
- The worksheet and exit ticket, one copy each per student
- The rubric, one copy per student if you grade the work

PREP

Make copies of the worksheet and exit ticket. Before class, open mytekdev.com/try/telescopelab?lesson=the-nights-queue-a-greedy-scheduler&ref=tpt-telescopelab-4 once on a student computer, which checks that your school's filter allows the app, and follow the short guide yourself: about ten minutes.

ASSESSMENT

- Formative: the questions under What to look for, asked while students work, and the exit ticket (answer key included).
- Summative: the worksheet, scored with the rubric.

TECHNOLOGY AND PRIVACY

Runs in any browser, Chromebooks included, with nothing to install. Students need no account and no email: the link works signed out, and nothing they do is stored on our server unless they choose to save.

Lesson sequence

MINUTES	STAGE	WHAT YOU DO	WHAT STUDENTS DO
0 to 5	**Warm-up** Hook	Show the title, essential question and Today I can slides (about 1 minute). Then the warm-up: seven things to do in one evening.	Answer the warm-up on the worksheet.
5 to 13	**Direct instruction (I do)** Model it	Project Level 8. A 10 meter class telescope on a summit in Hawaii, one night from 7:35 pm to 5:35 am, seven targets. Show the chart slide: each line is a target's height through the night. Read the starter in the Python tab: for target in scope.targets(), observe it. Ask what will happen, then Speed up three times and press play. Read the log aloud: five tries could not work, Jupiter and Uranus counted, 6 of 23, and the program ended at 9:55 pm with 7.7 hours of the night unused. **It's working when:** scope.priority(t), scope.needs(t) (minutes) and scope.minutes_left(t) (how long it stays above 30 degrees from now; 0 if it is not up) are what a choice can read. scope.done(t) says whether a block on it has counted.	Fill in the vocabulary and record the starter's result.
13 to 23	**Guided practice (We do)** List order, then greedy	Students open the link or scan the QR code. A short guide runs the starter, then has them type the scheduler from the worksheet over the for loop and play. Read it together first: a while loop; inside it, a for loop that keeps the best target so far; then observe it, or wait 600 seconds. It reaches 18 points at 1:40 am and 23 by dawn. **Watch for:** Indentation. The if best is None line lines up with for t in, not with the ifs inside it.	Follow the guide. Type the scheduler and record when it reaches 18 and what it does each hour.
23 to 37	**Independent practice (You do)** Take it apart	Students test four changed versions, one at a time, and record points and failed tries: (1) remove scope.minutes_left(t) >= scope.needs(t) and the and before it; (2) instead, put the check back and change if best is None or ... to if best is None; (3) wait 1800 instead of 600; (4) a single for loop that observes each target if it stays high long enough. Walk the room with the What to look for questions. **If they ask:** "Without the check, the scheduler tried M92 over and over. Why M92, and why did that cost the night?" The check **Watch for:** Students who conclude priority is useless. It did not change tonight's total; with checks every 30 minutes it changes the order (M104 before M51), and on a crowded night it decides what gets dropped.	Fill in the versions table and answer the questions.

MINUTES	STAGE	WHAT YOU DO	WHAT STUDENTS DO
37 to 45	**Closure** Which part did the work? And exit ticket	Show the closure slide and take answers. Then name the algorithm: greedy, the best choice possible now, repeated; fast and simple, and not always the best plan for the whole night. Hand out the exit ticket.	Discuss, then answer the exit ticket.

Before you teach: a ten-minute walkthrough

Open mytekdev.com/try/telescopelab?lesson=the-nights-queue-a-greedy-scheduler&ref=tpt-telescopelab-4. A short guide opens over Level 8, The night's queue, with the program in the Python tab. The chart shows each target's height through the night with the 30 degree line, and the log under it lists every block and every try that could not work. The night waits for the program: nothing moves until you press play. Speed up three times (1000 times faster) runs the 10 hour night in under a minute.

1. **The targets.** M92 (priority 5, 60 minutes), M104 the Sombrero (4, 60), M51 the Whirlpool (3, 60), M81 (3, 60), M66 (2, 45), Jupiter (2, 30) and Uranus (4, 60): 23 points. A block counts only if the target stays above 30 degrees in the dark for all of it; a try that cannot work costs 10 minutes.
2. **The starter.** For each target in the list's order, observe it. M92, M104, M51, M81 and M66 have not risen high enough (7:35 to 8:25 pm, 10 minutes each), then Jupiter (8:25 to 8:55) and Uranus (8:55 to 9:55) count. The program ends at 9:55 pm with 6 of 23.
3. **The scheduler.** Uranus at 7:35 pm, Jupiter 8:35, M81 9:05, M66 10:05, then waits until M51 is high enough at 11:40 pm, M104 at 12:40 am (18 points at 1:40 am, the level done), waits again, and M92 at 3:30 am: 23 by dawn, no failed tries.
4. **The versions.** (1) No check: it tries M92, the highest priority, every 10 minutes from 7:35 pm until it is high enough at 3:25 am (47 failed tries), then M92 and M104, 9 points; one more failed try on Uranus as dawn comes (48 in all), and dawn stops the program ("Dawn: the night is over"). (2) No priority comparison: 23, the same plan. (3) wait 1800: 23, M104 at 11:50 pm before M51, 18 at 1:50 am. (4) One for loop with the check: Uranus only, 4 points, and the program ends. With wait 60: 18 at 1:36 am. None of these change from run to run.

The program. Indent is how many levels in (four spaces each):

INDENT	LINE
0	from telescope import Telescope
0	scope = Telescope()
0	while True:
1	best = None
1	for t in scope.targets():
2	if not scope.done(t) and scope.minutes_left(t) >= scope.needs(t):
3	if best is None or scope.priority(t) > scope.priority(best):

INDENT	LINE
4	best = t
1	if best is None:
2	scope.wait(600)
1	else:
2	scope.observe(best)

> This is the program on the worksheet. A student who types it with the indents as shown reaches 23.

Background for the teacher

Real observatories plan every night: which targets, in what order, given when each is high enough, how dark it is, where the Moon is, and how much each program counts. A greedy scheduler, at each free moment the best target that can be done now, is simple and fast, and real schedulers start from it and add tweaks. It is not always best for the whole night: a greedy choice now can use time that a better plan would save for something that rises later.

Tonight's seven targets fit in the night with room to spare, as long as no time is wasted on tries that cannot work and the program comes back for targets that rise later. That is why the check and the wait do the work and the priority comparison changes only the order. The summit, the targets' heights (from Astronomy Engine) and the 30 degree limit are the app's model of a real 10 meter class telescope's night; the targets' positions come from OpenNGC.

Astronomers, telescope operators (the job this app is closest to), engineers who build cameras and adaptive optics, and data scientists who sort survey pictures all work with telescopes like these. Anyone can help with real data through citizen science projects such as Zooniverse and the AAVSO.

What to look for

Ask these while students work. A student who has understood the lesson can answer them without help:

- Why did the starter score only 6?
- What does scope.minutes_left(t) >= scope.needs(t) check, and why does the scheduler need it?
- Why does the scheduler need a wait, and what would happen without one?
- Why did removing the priority comparison change nothing tonight?
- Why is a while loop the right structure here and a for loop not?

> The fifth question is the one that matters. A student who says "a for loop goes through the list once, but targets rise later, so the program has to keep choosing until the night ends" can justify a control structure. A student who says "the while loop worked" has the result but not the idea.

Cross-curricular extensions

- Math: from the chart, estimate when each target rises above 30 degrees. Check your estimates against scope.minutes_left at those times.
- Earth science: why do the galaxies rise later in the night than Uranus sets? Use the idea that the sky turns 15 degrees an hour.
- Life skills: plan a Saturday with errands that are open only at certain times, greedily, then by planning ahead. Which is better?
- Careers: look up what a telescope scheduler or night assistant does at a large observatory.

Vocabulary

Greedy algorithm	An algorithm that makes the best choice it can at each step, without planning the whole night ahead.
Priority	How much a target counts: 1 to 5 points.
Constraint	A rule a choice must meet: here, above 30 degrees, in the dark, for the whole block.
Altitude	How high a target is above the horizon, in degrees.
Block	The time a target needs, 30 to 60 minutes, spent on it in one go.
Selection	Finding the best item in a list by comparing each with the best so far.

Differentiation

ENGLISH LEARNERS

- Keep the vocabulary slide up, and point at the telescope, the eyepiece view or the box at the top as you say each word: the picture carries the meaning.
- Give a sentence frame on a card: At ____ the scheduler chose ____ because it was up for ____ minutes and worth ____ points.
- Print the chart and have students mark, with a finger, which targets are above the line at 8, 11 and 3.

STUDENTS WITH IEPS OR 504 PLANS

- Pair students, one typing and one reading the program aloud.
- Give the scheduler as a file to paste instead of typing it.
- Where the IEP allows a modified assignment, score the first two rubric rows only.

STUDENTS WHO NEED MORE SUPPORT

- Type the scheduler together on the projector, one line at a time, saying what each does.
- Test only versions (1) and (4).
- The log under the chart lists every block and every failed try, with the time and the reason.

STUDENTS READY FOR MORE

- Make the choice smarter: among the targets that can be done now, prefer the one that sets soonest. Does it change the order or the time 18 is reached?
- Count how many times the program asked scope.minutes_left in the whole night. How could it ask less?

- Write a version that plans the whole night first, from the heights at each hour, then observes in that order. Is it better than greedy here?

Adapting for other grades

Younger students: In middle school, run the starter and the scheduler together on the board, then let pairs test version (1) only. Talk through the log rather than the code. Score the first two rubric rows, total 8, and 6 meets the objectives.

Older students: In AP Computer Science, write the selection as its own function, best_target(), and the scheduler as a loop that calls it. Compare greedy with trying every order (7 targets give 5,040 orders): why do real observatories use greedy schedulers, with tweaks, instead?

Student worksheet

Name: _____ Date: _____

Open the app: scan the QR code, or go to mytekdev.com/try/telescopelab?lesson=the-nights-queue-a-greedy-scheduler

Essential question: How does a program choose well, one step at a time?

Warm-up: Seven things to do in one evening, some only before 9 and some only after midnight: is the list's order a good plan? _____

The program. Indent is how many levels in (four spaces each):

INDENT	LINE
0	from telescope import Telescope
0	scope = Telescope()
0	while True:
1	best = None
1	for t in scope.targets():
2	if not scope.done(t) and scope.minutes_left(t) >= scope.needs(t):
3	if best is None or scope.priority(t) > scope.priority(best):
4	best = t
1	if best is None:
2	scope.wait(600)
1	else:
2	scope.observe(best)

1. **Vocabulary.** Greedy algorithm, priority, constraint, altitude, block, selection: write each in your own words.

   _____

   _____

   _____

   _____

2. **The starter.** Points: _____ of 23. Tries that could not work: _____. The program ended at _____.
3. **The scheduler.** Type the program above in place of the starter (its first two lines are the same). 18 points at _____. Points by dawn: _____. Failed tries: _____.

4. **Versions.** (1) No check: points _____ failed tries _____. (2) No priority comparison: _____ _____. (3) wait 1800: _____ _____, first after M66: _____. (4) One for loop with the check: _____ _____.

1. Which part of the rule did the work tonight, and how do your versions show it?

2. Why does the scheduler use a while loop, not a for loop over the targets?

Exit ticket

Name: _____

1. A greedy algorithm: (A) tries every possible order (B) makes the best choice it can at each step (C) takes the items in the order given

2. The starter scored 6 mainly because: (A) the targets were worth little (B) it tried targets before they rose and never came back (C) the night was too short

3. A friend's scheduler picks the highest priority target that is not done, with no other check. Use what you learned today to say what will happen, and what to add.

Telescope Lab: The Night's Queue, a Greedy Scheduler

Exit ticket

Name: _____

1. A greedy algorithm: (A) tries every possible order (B) makes the best choice it can at each step (C) takes the items in the order given

2. The starter scored 6 mainly because: (A) the targets were worth little (B) it tried targets before they rose and never came back (C) the night was too short

3. A friend's scheduler picks the highest priority target that is not done, with no other check. Use what you learned today to say what will happen, and what to add.

Answer key: exit ticket

QUESTION	ANSWER	WHY
1	B	At each free moment it picks the best target that can be done now.
2	B	Five targets had not risen; each failed try cost 10 minutes and the for loop went through the list once.
3	Open answer	It keeps trying M92 before it rises, wasting 10 minutes each time (47 tries on M92, 9 points by dawn). Add the check that the target stays above 30 degrees for the minutes it needs, and wait when nothing can be done.

Worksheet: what students should find

BLANK	WHAT THE APP SHOWS
The starter	6 of 23. 5 failed tries. Ended at 9:55 pm.
The scheduler	18 at 1:40 am. 23 by dawn. 0 failed tries.
Versions	(1) 9 points; 48 failed tries (47 on M92 from 7:35 pm until it rises at 3:25 am, then Uranus at dawn), and the program stops at dawn. (2) 23, 0. (3) 23, 0; after M66 comes M104. (4) 4, 0: Uranus only.
Question 1	The check and the wait: with them, any order of choosing gets 23 (version 2); without the check, priority alone gets 9.
Question 2	Targets rise later in the night, so the program must keep choosing until dawn; a for loop goes through the list once.

> These numbers are the same on every run and every computer. A student whose times differ by a few minutes changed the wait.

Rubric

Score each row from 1 to 4.

Name: _____

CRITERIA	4 EXCEEDS	3 MEETS	2 APPROACHING	1 BEGINNING
Writing the scheduler	Writes the scheduler, reaches 23, and explains each line, including why the while loop and the wait are needed.	Writes the scheduler and reaches 18 or more.	Writes part of it, or it does not run.	No scheduler.
Testing the parts	Tests all four versions, records points and failed tries for each, and names which part did the work tonight.	Tests all four versions and records each.	Tests one or two.	Does not test.
Justifying the structure	Explains why a while loop with a selection inside beats one for loop, with numbers, and when priority would matter.	Explains why the for loop scored less.	States the scheduler is better with no reason.	No explanation.

Total: ____ / 12. A score of 9 or more meets the objectives.

Standards

The lesson is built for CSTA 3A-AP-15: students justify a while loop with a selection inside over a single for loop, with the points and failed tries of five versions as evidence. Writing the scheduler from the greedy idea is 3A-AP-13. Greedy selection is a classic algorithm students adapt to the night's constraints, 3B-AP-10, a grades 11 and 12 standard.

FRAMEWORK	CODE	GRADES	STANDARD	HOW THE LESSON MEETS IT
CSTA 2017	**3A-AP-13**	9-10	Create prototypes that use algorithms to solve computational problems by leveraging prior student knowledge and personal interests.	Students turn the warm-up's idea, do what can be done now, into a working scheduler that scores 23 of 23.
CSTA 2017	**3A-AP-15**	9-10	Justify the selection of specific control structures when tradeoffs involve implementation, readability, and program performance, and explain the benefits and drawbacks of choices made.	Students compare a single for loop (4 or 6 points) with a while loop that selects the best target each time (23), and remove one part at a time to see what each costs.
CSTA 2017	**3B-AP-10**	11-12	Use and adapt classic algorithms to solve computational problems.	Students adapt a greedy selection, the best item by priority among those meeting a constraint, to scheduling a night, and test when its choice of order matters.

Checked November 2026 against: CSTA K-12 Computer Science Standards, Revised 2017.

Optional: save student work in a free class

The whole lesson runs at the link on the worksheet, with no accounts and no sign-up.

To keep students' work, and see it all in one place, make a free class. It takes about ten minutes, once, and covers every lesson in this series: mytekdev.com/tools/class-setup

Students in a class open mytekdev.com/tools/telescope-lab/lessons/the-nights-queue-a-greedy-scheduler/start instead of the worksheet link. Each one gets their own copy of the starter, and the same link reopens it next time.

Terms of use

Thank you for downloading this lesson. The plan, the handouts and the art were made by mytekDev.

You may:

- print and copy it for your students
- change it for your own classes
- post it on your class website or learning management system
- share this file with other teachers, whole and unchanged, with this page included

You may not:

- sell it, or put it in a product you sell
- share a changed copy
- claim it as your own

The app this lesson uses is free and runs in the browser. Students need no account. It is at mytekdev.com.

Questions, or found a mistake? Ask on the Teachers Pay Teachers Q&A for this resource, or write to us at mytekdev.com/contact.

© 2026 mytekDev. All rights reserved.