All Problems (Lecture and Practice)

● Easiest | ■ Medium | ◆ Hardest

#	Description	Length	Difficulty	Code Files	Requires Lecture
BSL P1	Write more arithmetic expressions.	5 min.	●	more-arithmetic-expression-starter.rkt more-arithmetic-expression-solution.rkt	exprs
BSL P2	Write out the step-by-step evaluation of expressions involving calls to primitives.	8 min.	■	evaluation-prims-starter.rkt evaluation-prims-solution.rkt	eval
BSL P3	Create an image using image composition primitives.	5 min.	●	tile-starter.rkt tile-solution.rkt	strs-imgs
BSL P4	Write expressions to operate on strings using primitives.	8 min.	■	glue-starter.rkt glue-solution.rkt	strs-imgs
BSL P5	Write expressions to operate on booleans using primitives.	7 min.	●	compare-images-starter.rkt compare-images-solution.rkt	bools-if-exprs
BSL P6	Step by step evaluation of a call to a function that calls a number primitive in its body.	7 min.	●	more-foo-evaluation-starter.rkt more-foo-evaluation-solution.rkt	stepper
BSL P7	Step by step evaluation of a call to a function that calls a string primitive in its body.	7 min.	●	even-more-foo-evaluation-starter.rkt even-more-foo-evaluation-solution.rkt	stepper
BSL P8	Write expression to produce background image of Canadian flag.	5 min.	●	cflag-starter.rkt cflag-solution.rkt	strs-imgs
BSL P9	Write even more arithmetic expressions.	5 min.	●	even-more-arithmetic-expression-starter.rkt even-more-arithmetic-expression-solution.rkt	exprs
BSL P10	Write an expression that operates on images using image primitives.	5 min.	●	triangle-starter.rkt triangle-solution.rkt	strs-imgs
BSL P11	Step by step evaluation of a call to a function that has an if expression in its body.	7 min.	●	bobble-evaluation-starter.rkt bobble-evaluation-solution.rkt	stepper
BSL P12	Fix the error(s) in a call to an image primitive.	5 min.	●	debug-rectangle-starter.rkt debug-rectangle-solution.rkt	strs-imgs

#	Description	Length	Difficulty	Code Files	Requires Lecture
BSL P13	Write an arithmetic expression.	5 min.	🟢	arithmetic-expression-starter.rkt arithmetic-expression-solution.rkt	exprs
BSL P14	Write an expression that operates on images using image primitives.	10 min.	🟢	overlay-starter.rkt overlay-solution.rkt	strs-imgs
BSL P15	Write a function that produces the larger of two given numbers.	5 min.	🟢	function-writing-starter.rkt function-writing-solution.rkt	fun-def
BSL P16	Step by step evaluation of a call to a function that has an if expression in its body.	15 min.	🟦	foo-evaluation-starter.rkt foo-evaluation-solution.rkt	stepper
HtDF L1	Design a function to yell!	7 min.	🟢	yell-starter.rkt yell-solution.rkt	yell
HtDF L2	Design a function to calculate the area of a square.	8 min.	🟢	area-starter.rkt area-solution.rkt	area
HtDF L3	Design a function to calculate the area of an image.	8 min.	🟢	image-area-starter.rkt image-area-solution.rkt	img-area
HtDF L4	Design a function to determine if an image is tall.	10 min.	🟢	tall-starter.rkt tall-solution.rkt	tall
HtDF P1	Design a function to summon items.	10 min.	🟢	summon-starter.rkt summon-solution.rkt	full
HtDF P2	Design a function to check if length of a string is less than 5.	10 min.	🟢	less-than-five-starter.rkt less-than-five-solution.rkt	full
HtDF P3	Design a function to put a box around a given image.	15 min.	🟦	boxify-starter.rkt boxify-solution.rkt	full
HtDF P4	Write more than one stub for a function given its signature and purpose.	5 min.	🟢	pluralize-stubs-starter.rkt pluralize-stubs-solution.rkt	full
HtDF P5	Design a function to produce a blue solid triangle of a given size.	10 min.	🟢	blue-triangle-starter.rkt blue-triangle-solution.rkt	full
HtDF P6	Fix the error(s) in a function that doubles a given number.	7 min.	🟢	double-error-starter.rkt double-error-solution.rkt	full
HtDF P7	Design a function to create a square based on the given colour.	10 min.	🟢	make-box-starter.rkt make-box-solution.rkt	full
HtDF P8	Design a function to produce a question by adding "?" to the end of a given string.	10 min.	🟦	ensure-question-starter.rkt ensure-question-solution.rkt	full
HtDF P9	Design a function to calculate the distance between two points.	20 min.	◆	cartesian-starter.rkt cartesian-solution.rkt	full

#	Description	Length	Difficulty	Code Files	Requires Lecture
HtDD L1	Design a data definition for all the citys in the world.	8 min.	🟢	city-name-starter.rkt city-name-solution.rkt	atomic
HtDD L2	Given the data definition for City, design a function that checks if a given city is the best city in the world.	8 min.	🟢	best-starter.rkt best-solution.rkt	atomic-htdf
HtDD L3	Design a data definition for seat numbers in a theatre.	8 min.	🟢	seat-num-starter.rkt seat-num-solution.rkt	interv
HtDD L4	Design a data definition to represent a student's grade.	10 min.	🟢	letter-grade-starter.rkt letter-grade-solution.rkt	enum
HtDD L5	Design a data definition to represent the current state of a New Year's Eve countdown.	18 min.	🟦	countdown-starter.rkt countdown-solution.rkt	itemz
HtDD L6	Given the data definition for SeatNum, design a function that determines if the seat number is on the aisle.	10 min.	🟢	aisle-starter.rkt aisle-solution.rkt	interv-htdf
HtDD L7	Given the data definition for LetterGrade, design a function that produces the next highest grade.	10 min.	🟢	bump-up-starter.rkt bump-up-solution.rkt	enum-htdf
HtDD L8	Given the data definition for Countdown, design a function that produces an image of the current state.	15 min.	🟦	countdown-to-display-starter.rkt countdown-to-display-solution.rkt	itemz-htdf
HtDD P1	Design a data definition and a function for a program to track a ski lodge's payroll.	15 min.	🟢	employees-starter.rkt employees-solution.rkt	interv-htdf
HtDD P2	Design a data definition and a function for a program to classify buildings in downtown Vancouver.	20 min.	🟦	demolish-starter.rkt demolish-solution.rkt	enum-htdf
HtDD P3	Design a data definition and a function for a program to track a rocket's return to Earth.	25 min.	◆	rocket-starter.rkt rocket-solution.rkt	itemz-htdf
HtDD P4	Fix error(s) in the revised data definition for LetterGrade.	8 min.	🟢	letter-grade-error-starter.rkt letter-grade-error-solution.rkt	enum
HtDD P5	Discuss the importance of following style rules and fix a given data definition to follow them.	8 min.	🟢	style-rules-starter.rkt style-rules-solution.rkt	interv
HtDD P6	Design a data definition and a function to determine whether a bike route is exclusively designated for bicycles.	20 min.	🟦	bike-route-starter.rkt bike-route-solution.rkt	enum-htdf
HtDD P7	Fix the error(s) in a function we have developed for the Rocket program.	7 min.	🟢	rocket-error-starter.rkt rocket-error-solution.rkt	itemz-htdf
HtDD P8	Design a function based on a given data definition for compass directions.	8 min.	🟢	direction-starter.rkt direction-solution.rkt	enum-htdf
HtDD P9	Design a data definition to represent airline dinners, and a function to let flight attendants know a passenger's choice.	18 min.	🟦	dinner-starter.rkt dinner-solution.rkt	enum-htdf

#	Description	Length	Difficulty	Code Files	Requires Lecture
HtDW L1	A cat that moves across the screen.	40 min.	🟦	cat-starter.rkt cat-v3.rkt	da, main, wish-list
HtDW L2	Given an existing world program, add a mouse handler to reset the position of the moving cat.	25 min.	🟦	add-mouse-handler-starter.rkt cat-v4.rkt	add-key
HtDW P1	Design an animation of a countdown.	30 min.	🟢	countdown-animation-starter.rkt countdown-animation-solution.rkt	da, main, wish-list
HtDW P2	Design an animation of a traffic light.	50 min.	🟦	traffic-light-starter.rkt traffic-light-solution.rkt	da, main, wish-list
Compound L3	A cow that wanders back and forth across the screen.	120 min.	🟦	cowabunga-starter.rkt cowabunga-v0.rkt cowabunga-v1.rkt cowabunga-v2.rkt cowabunga-v3.rkt cowabunga-v4.rkt cowabunga-v5.rkt cowabunga-v6.rkt	htdw
Compound P1	Design a compound data definition to represent movies, and a function to compare their release dates.	25 min.	🟦	movie-starter.rkt movie-solution.rkt	dd
Compound P2	Design a world to represent a growing and rotating red box.	80 min.	◆	spinning-starter.rkt spinning-solution.rkt	htdw
Compound P3	Design a compound data definition to represent students, and a function to monitor their allergies.	25 min.	🟦	student-starter.rkt student-solution.rkt	dd
Compound P4	Design a compound data definition to represent trips, and a function to compare their lengths.	25 min.	🟦	trip-starter.rkt trip-solution.rkt	dd
Compound P5	Design a world to represent grass that grows and is replanted.	80 min.	◆	growing-grass-starter.rkt growing-grass-solution.rkt	htdw
Compound P6	Design a world where the mouse position is displayed at the mouse cursor.	50 min.	🟦	tracker-starter.rkt tracker-solution.rkt	htdw
Compound P7	Design a world where a lambda rolls back and forth across the screen.	100 min.	◆	rolling-lambda-starter.rkt rolling-lambda-solution.rkt	htdw
Compound P8	Write down the evaluation steps for an expression that has compound data.	5 min.	🟢	compound-evaluation-starter.rkt compound-evaluation-solution.rkt	dd
Compound P9	Design an animation of throwing a water balloon.	90 min.	◆	water-balloon-starter.rkt water-balloon-solution.rkt	htdw

#	Description	Length	Difficulty	Code Files	Requires Lecture
Compound P10	Design a simple one-line text editor.	90 min.	◆	simple-text-editor-starter.rkt simple-text-editor-solution.rkt	htdw
Self-Ref P1	Design a function to calculate the total number of individual characters in a list of strings.	18 min.	🟢	total-string-length-starter.rkt total-string-length-solution.rkt	list-fun
Self-Ref P2	Design a function to double every number in a list.	18 min.	🟢	double-all-starter.rkt double-all-solution.rkt	des-w-list
Self-Ref P3	Design a data definition to represent a list of booleans, and a function to determine if all values in a given list are true.	35 min.	🟦	boolean-list-starter.rkt boolean-list-solution.rkt	des-w-list
Self-Ref P4	Design a function to add '!' to each string in a list of strings.	18 min.	🟢	yell-all-starter.rkt yell-all-solution.rkt	list-fun
Self-Ref P5	Design a function to find the largest number in a list of numbers.	18 min.	🟢	largest-starter.rkt largest-solution.rkt	des-w-list
Self-Ref P6	Design a data definition to represent a list of images, and a function to find the sum of areas from a list of images.	35 min.	🟦	image-list-starter.rkt image-list-solution.rkt	des-w-list
Ref L1	A tuition graphing program and intermediate solutions.	60 min.	◆	tuition-graph-starter.rkt tuition-graph-v1.rkt tuition-graph-v2.rkt tuition-graph-v3.rkt tuition-graph-v4.rkt tuition-graph-v5.rkt tuition-graph-v6.rkt	part-1-2-3
Ref P1	Design the tuition graph bar chart function based on an alternative data definition for School.	50 min.	◆	alternative-tuition-graph-starter.rkt alternative-tuition-graph-solution.rkt	part-3
Ref P2	Design a world program that has an arbitrary number of spinning bears.	100 min.	◆	spinning-bears-starter.rkt spinning-bears-solution.rkt	part-3
Ref P3	Design a function to find the lowest tuition, and a function to produce a list of school names.	45 min.	◆	tuition-graph-c-starter.rkt tuition-graph-c-solution.rkt	part-3
Naturals L1	Design functions that operate on the Natural data definition.	25 min.	🟢	naturals-starter.rkt naturals-solution.rkt	nat nums
Naturals L2	Design functions that does arithmetic operations on the NATURAL data definition.	50 min.	🟦	new-numerals-starter.rkt new-numerals-solution.rkt	parlor
Naturals P1	Design a function that computes the sum of all natural numbers from zero to n.	10 min.	🟢	sum-to-n-starter.rkt sum-to-n-solution.rkt	nat nums

#	Description	Length	Difficulty	Code Files	Requires Lecture
Naturals P2	Design a function that produces an image of the numbers from n to zero, side by side.	15 min.	🟦	decreasing-image-starter.rkt decreasing-image-solution.rkt	nat nums
Naturals P3	Design a function that produces a list of all the odd numbers from n to zero.	15 min.	🟦	odd-from-n-starter.rkt odd-from-n-solution.rkt	nat nums
Naturals P4	Design a function that produces an image of n concentric circles of a given colour.	15 min.	🟦	concentric-circles-starter.rkt concentric-circles-solution.rkt	nat nums
Helpers L1	Creating a program to sort a list of images and lay them out next to each other, as demonstrated in lecture, with all intermediate solutions.	60 min.	🟦	arrange-images-starter.rkt arrange-images-v1.rkt arrange-images-v2.rkt arrange-images-v3.rkt arrange-images-v4.rkt arrange-images-v5.rkt arrange-images-v6.rkt	all
Helpers P1	Design a program to render a dodgeball game roster.	100 min.	🟦	render-roster-starter.rkt render-roster-solution.rkt	all
Helpers P2	Design a world program to make it rain where you want it to. In this version of the program you should remove drops that have already fallen off the screen from the world state. Be sure to use all appropriate helper function rules. There are a total of 9 functions in our solution, so this problem takes some time.	120 min.	🟦	making-rain-filtered-starter.rkt making-rain-filtered-solution.rkt	all
BSTs L1	Creating a program to find an account in a list of account given its account number.	15 min.	🟢	lookup-in-list-starter.rkt lookup-in-list-solution.rkt	loa
BSTs L2	Creating a data definition for a Binary Search Tree.	25 min.	🟦	bst-dd-starter.rkt bst-dd-solution.rkt	dd
BSTs L3	Creating a function that searches through a BST.	25 min.	🟦	lookup-in-bst-starter.rkt lookup-in-bst-solution.rkt	lookup
BSTs L4	Creating a function that renders a BST	30 min.	🟦	render-bst-starter.rkt render-bst-solution.rkt	render
BSTs P1	Design a function to count the number of nodes in a BST.	7 min.	🟢	count-nodes-starter.rkt count-nodes-solution.rkt	lookup
BSTs P2	Design a function that sums the keys in a BST.	7 min.	🟢	sum-keys-starter.rkt sum-keys-solution.rkt	lookup
BSTs P3	Design a function to determine the height of a BST.	20 min.	🟦	height-starter.rkt height-solution.rkt	lookup
BSTs P4	Design a function that inserts a node in the proper place in a BST.	30 min.	🟦	insert-starter.rkt insert-solution.rkt	lookup

#	Description	Length	Difficulty	Code Files	Requires Lecture
BSTs P5	Design functions to determine whether a BST is balanced.	30 min.	🟦	balance-factor-starter.rkt balance-factor-solution.rkt	lookup
BSTs P6	Design a function that renders a BST with lines.	45 min.	◆	render-bst-w-lines-starter.rkt render-bst-w-lines-solution.rkt	render
Mutual-Ref L1	An implementation of a file system, as demonstrated in lecture, with all intermediate solutions.	60 min.	◆	fs-starter.rkt fs-v1.rkt fs-v2.rkt fs-v3.rkt fs-v4.rkt	all
Mutual-Ref P1	Several function design problems for arbitrary-arity trees.	90 min.	🟦	image-organizer-starter.rkt image-organizer-solution.rkt	fun-part-2
Mutual-Ref P2	Design a function to find a person in a person tree.	30 min.	🟦	find-person-starter.rkt find-person-solution.rkt	fun-part-2
Mutual-Ref P3	Represent information about descendant family trees from Harry Potter and design functions that operate on them.	90 min.	🟦	hp-family-tree-starter.rkt hp-family-tree-solution.rkt	fun-part-2
2-One-Of P1	Design a function that concatenates two lists.	25 min.	🟦	concat-starter.rkt concat-solution.rkt	code
2-One-Of P2	Design a function that merges two sorted lists into a single sorted list.	35 min.	🟦	merge-starter.rkt merge-solution.rkt	code
2-One-Of P3	Design a function that produces a list of the corresponding elements of the given two lists.	30 min.	🟦	zip-starter.rkt zip-solution.rkt	code
2-One-Of P4	Design a function that determines if a sequence of characters matches a given pattern.	40 min.	🟦	pattern-match-starter.rkt pattern-match-solution.rkt	code
2-One-Of P5	Represent and operate on information about USA Ultimate championship bracket.	40 min.	🟦	championship-bracket-starter.rkt championship-bracket-solution.rkt	code
Local L1	Refactoring of file system developed in Mutual-Ref.	25 min.	🟦	fs-v4.rkt fs-v5.rkt fs-v6.rkt fs-v7.rkt	encap, recomp
Local P1	Step by step evaluation of a call to a function that involves local.	15 min.	🟢	evaluate-boo-starter.rkt evaluate-boo-solution.rkt	eval
Local P2	Improve the performance of a function that produces a simple rendering of a bst with lines.	30 min.	🟦	render-bst-w-lines-faster-starter.rkt render-bst-w-lines-faster-solution.rkt	avoid-recomp
Local P3	Step by step evaluation of a call to a function that involves local.	30 min.	🟦	evaluate-foo-starter.rkt evaluate-foo-solution.rkt	eval

#	Description	Length	Difficulty	Code Files	Requires Lecture
Local P4	Encapsulate the total-area function into local.	25 min.	●	encapsulate-total-area-starter.rkt encapsulate-total-area-solution.rkt	encap
Local P5	Improve a function that determines which team knocked a given team out of the tournament.	30 min.	■	championship-bracket-improved-starter.rkt championship-bracket-improved-solution.rkt	encap, avoid-recomp
Abstraction L1	Introduction to abstracting functions, as demonstrated in lecture, with intermediate solutions.	45 min.	■	parameterization-starter.rkt parameterization-v2.rkt parameterization-v3.rkt	from-ex
Abstraction L2	Design functions that use built-in abstract list functions.	25 min.	●	using-built-ins-starter.rkt using-built-ins-solution.rkt	built-in
Abstraction L3	Design functions that use built-in abstract list functions with closure.	35 min.	■	closures-starter.rkt closures-solution.rkt	closures
Abstraction P1	Design a function to produce only wide images from a list of images.	15 min.	●	wide-only-starter.rkt wide-only-solution.rkt	built-in
Abstraction P2	Design a function to produce all the favourited photos in the given album.	25 min.	■	photos-starter.rkt photos-solution.rkt	built-in
Abstraction P3	Design an abstract function to simplify the sum-of functions.	30 min.	●	abstract-sum-starter.rkt abstract-sum-solution.rkt	from-ex-3
Abstraction P4	Design an abstract function to simplify the some functions.	40 min.	■	abstract-some-starter.rkt abstract-some-solution.rkt	from-ex-3
Abstraction P5	Write expressions that uses abstract functions to produce a list of ellipses and images of ellipses.	25 min.	■	ellipses-starter.rkt ellipses-solution.rkt	built-in
Abstraction P6	Design a function to produce the linear lengths of a list of bags.	25 min.	●	bag-starter.rkt bag-solution.rkt	built-in
Abstraction P7	Design a function to produce the sum of the first n odd numbers.	25 min.	■	sum-n-starter.rkt sum-n-solution.rkt	built-in
Abstraction P8	Design an abstract fold function for Dir and some functions that uses this abstract function.	60 min.	◆	fold-dir-starter.rkt fold-dir-solution.rkt	fold
Abstraction P9	Design abstract functions to simplify given functions.	45 min.	■	accounts-starter.rkt accounts-solution.rkt	from-ex-3
Abstraction P10	Use built-in abstract functions to design a function that produces the sum total of rainfall on warm days.	30 min.	■	weather-starter.rkt weather.ss weather-solution.rkt	built-in

#	Description	Length	Difficulty	Code Files	Requires Lecture
Genrec L1	Design of a few basic fractals, as demonstrated in lecture, with intermediate solutions.	30 min.	🟢	fractals-starter.rkt fractals-v1.rkt fractals-v2.rkt	fractals
Genrec L2	Design three part termination argument for generative recursions.	45 min.	🟦	termination-starter.rkt termination-solution.rkt	term-arg
Genrec P1	Design a function to draw a circle fractal.	90 min.	◆	circle-fractal-starter.rkt circle-fractal-solution.rkt	term-arg
Genrec P2	Design a function to draw a Van Koch fractal line. The GEOMETRY of this problem is much more difficult than the triangle and the carpet. If you feel confident with cartesian geometry then this function may not be too difficult. If you feel less confident then you may find this problem to be too difficult.	110 min.	◆	van-koch-starter.rkt van-koch-solution.rkt	term-arg
Genrec P3	Design a world program to create the cantor set.	90 min.	🟦	cantor-starter.rkt cantor-solution.rkt	term-arg
Search L1	Design a program that solves a sudoku board, as demonstrated in lecture, with intermediate solutions.	180 min.	◆	sudoku-starter.rkt sudoku-v1.rkt sudoku-v2.rkt sudoku-v3.rkt sudoku-v4.rkt sudoku-solution.rkt sudoku-constraints.rkt	sudoku
Search P1	Design a program that tries to find the path through a maze.	120 min.	◆	maze-2w-starter.rkt maze-2w-v1.rkt maze-2w-solution.rkt	sudoku
Search P2	Design a program that solves triangle solitaire puzzles.	120 min.	◆	triangle-solitaire-starter.rkt triangle-solitaire-v2.rkt triangle-solitaire-v3.rkt triangle-solitaire-v4.rkt	sudoku
Search P3	Design a program that solves the n queens problem.	120 min.	◆	nqueens-starter.rkt nqueens-v1.rkt nqueens-v2.rkt nqueens-solution.rkt	sudoku
Accumulators P1	Design a function that drops every nth element from a list.	15 min.	🟢	dropn-starter.rkt dropn-solution.rkt	context-preserv-acc
Accumulators P2	Design a function that replicates elements in a list.	15 min.	🟢	replicate-elm-starter.rkt replicate-elm-solution.rkt	context-preserv-acc
Accumulators P3	Design a function that produces true if numbers in a list are strictly decreasing.	20 min.	🟢	strictly-decreasing-starter.rkt strictly-decreasing-solution.rkt	context-preserv-acc
Accumulators P4	Design a tail-recursive function that produces the average of numbers in a list.	15 min.	🟢	average-tr-starter.rkt average-tr-solution.rkt	tail-rec
Accumulators P5	Design a tail-recursive version of a function that produces the product of numbers in a list.	15 min.	🟢	product-tr-starter.rkt product-tr-solution.rkt	tail-rec

#	Description	Length	Difficulty	Code Files	Requires Lecture
Accumulators P6	Design a tail-recursive version of a function that produces the sum of naturals in [0, n].	15 min.	🟢	sum-n-tr-starter.rkt sum-n-tr-solution.rkt	tail-rec
Accumulators P7	Design a tail-recursive version of a function that produces the sum of odd numbers in a list.	15 min.	🟢	sum-odds-tr-starter.rkt sum-odds-tr-solution.rkt	tail-rec
Accumulators P8	Design a tail-recursive version of a function that produces a list of naturals from 1 to n.	15 min.	🟢	to-list-tr-starter.rkt to-list-tr-solution.rkt	tail-rec
Accumulators P9	Design a tail-recursive function that produces separate counts of odd and even numbers in a list.	20 min.	🟦	count-odd-even-tr-starter.rkt count-odd-even-tr-solution.rkt	tail-rec
Accumulators P10	Design a tail-recursive function that produces a list of the same elements in the opposite order.	20 min.	🟦	rev-tr-starter.rkt rev-tr-solution.rkt	tail-rec
Accumulators P11	Design functions that operate on house paths.	45 min.	🟦	house-path-starter.rkt house-path-solution.rkt	worklist-acc
Accumulators P12	Design a tail-recursive function that determines whether a binary tree contains a given key.	40 min.	🟦	contains-key-tr-starter.rkt contains-key-tr-solution.rkt	worklist-acc
Accumulators L1	Problems where an accumulator is needed to preserve information lost in the structural recursion template.	60 min.	🟦	skip1-starter.rkt skip1-solution.rkt skip1-v1.rkt skipn-starter.rkt skipn-solution.rkt skipn-v1.rkt	context-preserv-acc
Accumulators L2	Using accumulators to make functions tail recursive.	50 min.	🟦	sum-tr-starter.rkt sum-tr-solution.rkt	tail-rec
Accumulators L3	Making functions operating on an arbitrary-arity tree tail recursive using context preserving and worklist accumulators.	120 min.	◆	same-house-as-parent-v1.rkt same-house-as-parent-v2.rkt same-house-as-parent-v3.rkt same-house-as-parent-v4.rkt same-house-as-parent-v5.rkt same-house-as-parent-v6.rkt same-house-as-parent-v7.rkt same-house-as-parent-solution.rkt	worklist-acc
Graphs P1	Design a function that produces rooms reachable from a given room.	20 min.	🟢	all-reachable-starter.rkt all-reachable-solution.rkt	graphs-reachable
Graphs P2	Design a function that produces the number of rooms reachable from a given room.	20 min.	🟢	count-rooms-starter.rkt count-rooms-solution.rkt	graphs-reachable
Graphs P3	Design a function that produces a room with the given name.	20 min.	🟢	lookup-room-starter.rkt lookup-room-solution.rkt	graphs-reachable
Graphs P4	Design a function that produces the room with the most exits.	40 min.	🟦	max-exits-from-starter.rkt max-exits-from-solution.rkt	graphs-reachable

#	Description	Length	Difficulty	Code Files	Requires Lecture
Graphs P5	Design a function that produces a room to which the greatest number of other rooms have exits.	60 min.	◆	max-exits-to-starter.rkt max-exits-to-solution.rkt	graphs-reachable
Graphs L1	Walking through the design of a data definition for representing graphs, and a simple function that operates on those graphs.	120 min.	◆	graphs-v1.rkt graphs-v2.rkt graphs-v3.rkt graphs-v4.rkt graphs-v5.rkt	cyclic-data, templating, reachable