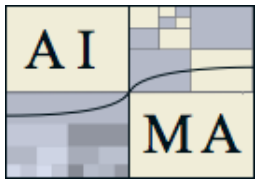
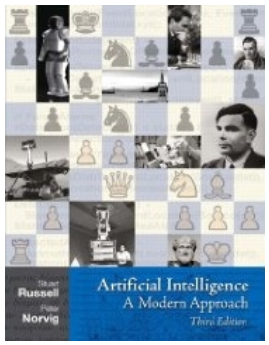


AIMA Home
Code
Contents
Courses
Errata
Instructors

Search AIMA



Online Code Repository

The goal is to have working code for all the algorithms in the book in a variety of languages. So far, we have Java, Lisp and Python versions of most of the algorithms. There is also some old code in C++, C# and Prolog, but these are not being maintained. We also have a directory full of data files. Let peter@norvig.com know what languages you'd like to see, and if you're willing to help.

Supported Implementations

We offer the followinglanguage choices, plus a selection of data that works with all the implementations:

- **Overall:** aimacode project on Github.
- **Java:** aima-java project, by Ravi Mohan and Ciaran O'Reilly and other contributors.
- **Python:** aima-python project, by Peter Norvig and many contributors. Read Me file.
- **Lisp:** aima-lisp, by Stuart Russell and Peter Norvig.

Unsupported Implementations

	AIMA Prolog	AIMA C++	AIMA C#
Maintainers	Larry Holder	Larry Holder	Kris Noesgaard
Years Developed	1995-96	1995-96	2005-2007
AIMA Code Overview	Prolog Overview	C++ Overview	C# Overview
Download	Prolog download	C++ download	C# download

Implementation Choices

What languages are instructors recommending? To get an approximate idea, I gave the query `norvig russell "Modern Approach"` along with the names of various languages and looked at the estimated counts of results on various dates:

Language	23 Sep 2004	2 Feb 2005	15 Jun 2007	6 Jan 2010
none	8,080	20,100	75,200	150,000
java	1,990	4,930	44,200	37,000
c++	875	1,820	35,300	105,000
lisp	844	974	30,100	19,000
prolog	789	2,010	23,200	17,000
python	785	1,240	18,400	11,000

Of course, neither recall nor precision is perfect for these queries, nor is the estimated number of results guaranteed to be accurate, but they offer a rough estimate of popularity. Also, the links in the table let you investigate individual courses using each language.

Index of Pseudocode in Book

Fig.	Page	Name (in Book)	Kind

Index of Programming Exercises in Book

Ex.	Page	Description

2	32	Environment	type
2.1	33	Agent	fn
2.3	34	Table-Driven-Vacuum-Agent	fn
2.7	45	Table-Driven-Agent	fn
2.8	46	Reflex-Vacuum-Agent	fn
2.10	47	Simple-Reflex-Agent	fn
2.12	49	Reflex-Agent-With-State	fn
3.1	61	Simple-Problem-Solving-Agent	fn
3	62	Problem	type
3.2	63	Romania	data
3	69	Node	type
3.7	70	Tree-Search	fn
3	71	Queue	type
3.9	72	Tree-Search	fn
3.13	77	Depth-Limited-Search	fn
3.14	79	Iterative-Deepening-Search	fn
3.19	83	Graph-Search	fn
4	95	Best-First-Search	fn
4	97	A*-Search	fn
4.5	102	Recursive-Best-First-Search	fn
4.11	112	Hill-Climbing	fn
4.14	116	Simulated-Annealing	fn
4.17	119	Genetic-Algorithm	fn
4.20	126	Online-DFS-Agent	fn
5	137	CSP	type
5.3	142	Backtracking-Search	fn
5.7	146	AC-3	fn
5.8	151	Min-Conflicts	fn
6.3	166	Minimax-Decision	fn
6.7	170	Alpha-Beta-Search	fn

2.7	57	Environment simulator for vacuum world (partially implemented)
3.9	90	Missionaries and Cannibals
3.10	91	Successor function for 8-puzzle
3.11	91	Iterative Lengthening Search
3.14	91	Path connecting two web pages
3.15-16	91	Shortest path in plane with polygonal obstacles
3.19	93	Search agents for vacuum world
4.4	134	Suboptimal admissible but inconsistent h(n)
4.8	135	Travelling Salesperson via Minimum Spanning Tree
4.10	135	Improved 8-puzzle heuristics
4.15	136	Local search to solve Travelling Salesperson
4.16	136	Compare hill-climbing, annealing on 8-puzzle, 8-queens
4.17	136	Hill-climbing for robot navigation
4.18	136	Compare A* and RBFS on 8-puzzle
5.7	159	Compare CSP algorithms on map-coloring
5.12	159	Find minimal cycle cutset
6.4	191	Move generator and evaluation functions for board games
6.6	191	Expectiminimax and *-alpha-beta for games with chance nodes
7.15	239	Extend PL-Wumpus-Agent to track all facts within the KB
8.11	269	Tell and Ask facts about family tre ein Fig. 8.5
8.17	270	Define addition for n-bit numbers; verify adder is correct.
9.14	317	Sorting in Prolog
9.15	318	Recursive rewrite rules (demodulators) in Logic programming
12.16	460	Modify And-Or-Graph-search to generate a cyclic plan
14.12	536	Relational Probabilistic Model of Soccer League
16.5	611	Determine utility of money for subjects
16.8	611	Model of airport-siting problem
17.6	647	Policy iteration on 4x3 (and larger) world
17.7	647	Find threshold values for R(s)
18.12	677	Decision-Tree-Learning with missing attribute values

7	195	KB	type
7.1	196	KB-Agent	fn
7.7	205	Propositional Logic Sentence	type
7.10	209	TT-Entails	fn
7	215	Convert to CNF	fn
7.12	216	PL-Resolution	fn
7.14	219	PL-FC-Entails?	fn
7.16	222	DPLL-Satisfiable?	fn
7.17	223	WalkSAT	fn
7.19	226	PL-Wumpus-Agent	fn
9	273	Subst	fn
9.1	278	Unify	fn
9.3	282	FOL-FC-Ask	fn
9.6	288	FOL-BC-Ask	fn
9.14	307	Otter	fn
11.2	380	Airport-problem	data
11.3	381	Spare-Tire-Problem	data
11.4	383	Three-Block-Tower	data
11	390	Partial-Order-Planner	fn
11.11	396	Cake-Problem	data
11.13	399	Graphplan	fn
11.15	403	SATPlan	fn
12.1	418	Job-Shop-Problem	data
12.3	421	Job-Shop-Problem-With-Resources	data
12.6	424	House-Building-Problem	data
12.10	435	And-Or-Graph-Search	fn
12.22	449	Continuous-POP-Agent	fn
12.23	450	Doubles-tennis	data
13.1	466	DT-Agent	fn
13	469	Discrete Probability Distribution	fn
13.4	477	Enumerate-Joint-Ask	fn
14.10	509	Elimination-Ask	fn
19.5	711	Inverse resolution in Logic Programming	
20.15	761	Perceptron and Decision-tree learning on a data set	
21.1	788	Compare algorithms for passive learning agent	
21.7	789	Exploring RL agent that uses direct utility estimation	
21.11	789	Two RL agents learning to play a game	
21.12	789	Reinforce and Pegasus algorithms for 4x3 world	
22.10	832	Chart-parsing with a packed tree representation	
22.11	832	Chart-parsing with a partial packed tree representation	
23.1	861	Stylometry: determining authorship	
23.2	861	Statistics of n-gram models	
23.3	861	Detection of spam email	
23.7	862	Regular expression or program to extract company names	
25.2	943	Monte Carlo localization	
25.4	943	Voronoi diagram	

14.12	512	Prior-Sample	fn
14.13	513	Rejection-Sampling	fn
14.14	515	Likelihood-Weighting	fn
14.15	517	MCMC-Ask	fn
15.4	546	Forward-Backward	fn
15.6	552	Fixed-Lag-Smoothing	fn
15.15	566	Particle-Filtering	fn
16.8	603	Information-Gathering-Agent	fn
17.4	621	Value-Iteration	fn
17.7	624	Policy-Iteration	fn
18.5	658	Decision-Tree-Learning	fn
18.10	667	AdaBoost	fn
18.14	672	Decision-List-Learning	fn
19.2	681	Current-Best-Learning	fn
19.3	683	Version-Space-Learning	fn
19.8	696	Minimal-Consistent-Det	fn
19.12	702	FOIL	fn
20.21	742	Perceptron-Learning	fn
20.25	746	Back-Prop-Learning	fn
21.2	768	Passive-ADP-Agent	fn
21.4	769	Passive-TD-Agent	fn
21.8	776	Q-Learning-Agent	fn
22.2	796	Naive-Communicating-Agent	fn
22.7	801	Chart-Parse	fn
23.1	837	Viterbi-Segmentation	fn
24.21	892	Align	fn

AI: A Modern Approach Modified: May 26, 2016