

## ALGORITHMIC THOUGHTS

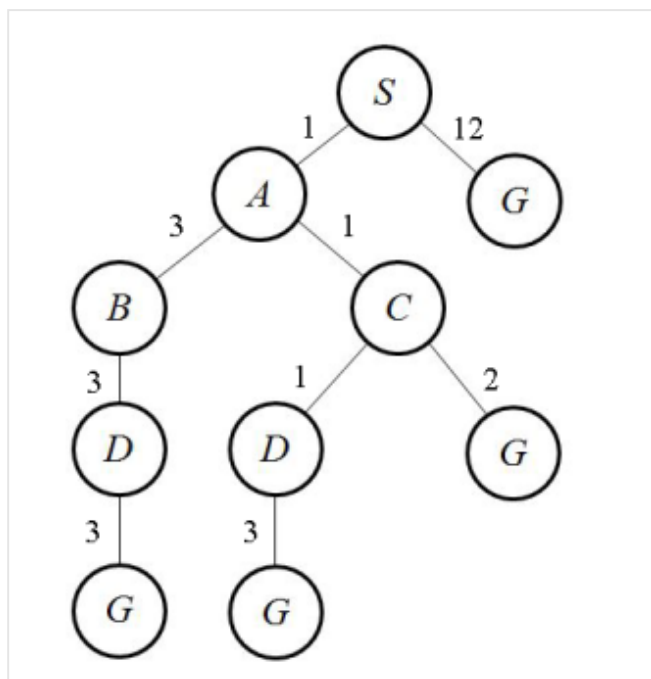
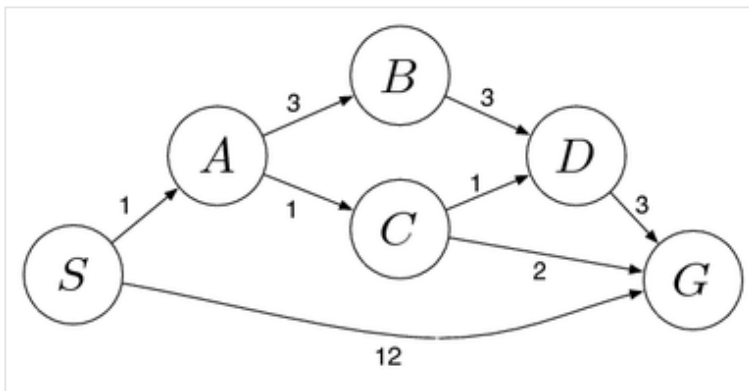
Artificial Intelligence | Machine Learning | Neuroscience | Computer Vision

[ABOUT](#)

# ARTIFICIAL INTELLIGENCE – A\* SEARCH ALGORITHM

January 4, 2013 · by Siddharth Agrawal · in *Artificial Intelligence* · 14 Comments

We will try to improve the efficiency of the Uniform Cost Search algorithm by using heuristics which we discussed in the previous post. By improving the efficiency I mean that the algorithm will expand less of the search tree and will give the optimal result faster. We start with the same search problem and search tree that we used for Uniform Cost Search. If you don't know what search problems are or how search trees are created, visit [this post](#).



Search this site...

### RECENT POSTS

- Machine Learning – Mini Batch K-Means
- Computer Vision – Shape Context Descriptor
- Machine Learning – DBSCAN
- Computer Vision – Feature Detection
- Artificial Intelligence – A\* Search Algorithm

### ARCHIVES

- July 2013
- June 2013
- May 2013
- January 2013
- December 2012
- November 2012

### BLOG STATS

- 120,106 hits

### CATEGORIES

- Artificial Intelligence
- Computer Vision
- Machine Learning

### BLOGROLL

- [Ameya's Positronic Brain AI](#)

Follow

## A\* Search Algorithm

We saw that Uniform Cost Search was optimal in terms of cost for a weighted graph. Now our aim will be to improve the efficiency of the algorithm with the help of heuristics. If you don't know what heuristics are, visit [this post](#). Particularly, we will be using admissible heuristics for A\* Search.

A\* Search also makes use of a priority queue just like Uniform Cost Search with the element stored being the path from the start state to a particular node, but the priority of an element is not the same. In Uniform Cost Search we used the actual cost of getting to a particular node from the start state as the priority. For A\*, we use the cost of getting to a node plus the heuristic at that point as the priority. Let  $n$  be a particular node, then we define  $g(n)$  as the cost of getting to the node from the start state and  $h(n)$  as the heuristic at that node. The priority thus is  $f(n) = g(n) + h(n)$ . The priority is maximum when the  $f(n)$  value is least. We use this priority queue in the following algorithm, which is quite similar to the Uniform Cost Search algorithm:

**Insert the root node into the queue**

**While the queue is not empty**

**Dequeue the element with the highest priority**

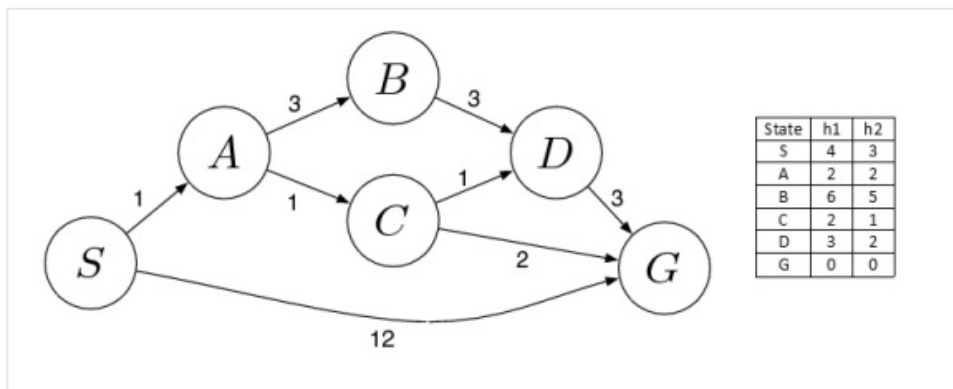
**(If priorities are same, alphabetically smaller path is chosen)**

**If the path is ending in the goal state, print the path and exit**

**Else**

**Insert all the children of the dequeued element, with  $f(n)$  as the priority**

Now let us apply the algorithm on the above search tree and see what it gives us. We will go through each iteration and look at the final output. Each element of the priority queue is written as  $[path, f(n)]$ . We will use  $h1$  as the heuristic, given in the diagram below.



**Initialization:**  $\{ [S, 4] \}$

**Iteration1:**  $\{ [S \rightarrow A, 3], [S \rightarrow G, 12] \}$

**Iteration2:**  $\{ [S \rightarrow A \rightarrow C, 4], [S \rightarrow A \rightarrow B, 10], [S \rightarrow G, 12] \}$

## Follow “Algorithmic Thoughts – Artificial Intelligence | Machine Learning | Neuroscience | Computer Vision”

Get every new post delivered to your Inbox.

Join 55 other followers



Build a website with WordPress.com

**Artificial  
Intelligence**  
Computer Vision Machine  
Learning

Follow

Iteration3: { [ S->A->C->G , 4 ] , [ S->A->C->D , 6 ] , [ S->A->B , 10 ] , [ S->G , 12 ] }

Iteration4 gives the final output as S->A->C->G.

Things worth mentioning:

->The creation of the tree is not a part of the algorithm. It is just for visualization.

->The algorithm returns the first path encountered. It does not search for all paths.

->The algorithm returns a path which is optimal in terms of cost, if an admissible heuristic is used(this can be proved).

The above example illustrates that A\* Search gives the optimal path faster than Uniform Cost Search. This is, however, true only if the heuristic is admissible. In general, the efficiency of the algorithm depends on the quality of the heuristic. The nearer the heuristic is to the actual cost, the better is the speed of the algorithm. Trivially, the heuristic can be taken to be 0, which gives the Uniform Cost Search algorithm.

With this, I have completed all the popular search algorithms in Artificial Intelligence. In the next post I will be writing about Feature Detection in Computer Vision.

## Follow “Algorithmic Thoughts – Artificial Intelligence | Machine Learning | Neuroscience | Computer Vision”

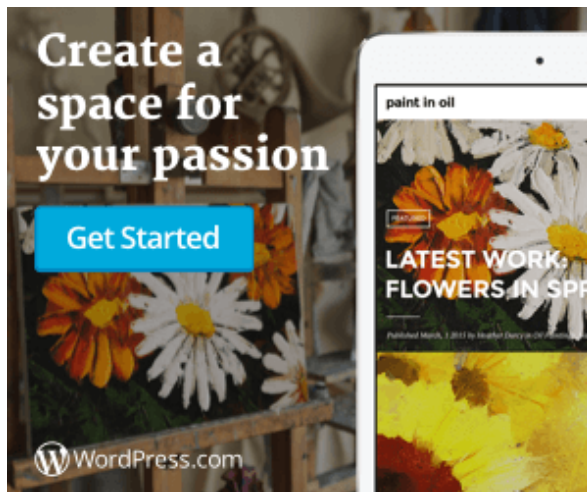
Get every new post delivered to  
your Inbox.

Join 55 other followers

Sign me up

Build a website with WordPress.com

About these ads



Share this:



Twitter



Facebook 2



Like



One blogger likes this.

Tags: a star, a star search, a\*, a\* search, a\* search algorithm, admissible, admissible heuristic, ai, algorithm, artificial intelligence, cost, easy, explanation, graph, heuristic, heuristics, path, priority queue, problem, search, search heuristic, search heuristics, simple, trees

Follow

## 14 comments



door hangers cheap · August 5, 2013 - 6:50 pm · [Reply](#)→

I do agree with all of the concepts you have presented in your post. They're really convincing and can definitely work. Nonetheless, the posts are very short for novices. May you please extend them a bit from subsequent time? Thanks for the post.



Siddharth Agrawal · August 5, 2013 - 10:02 pm · [Reply](#)→

Thanks for the comment. It would be great if you could tell me what details you are looking for, I will definitely extend as required.



Dan · October 7, 2013 - 4:12 am · [Reply](#)→

I've used several of the entries on your site to help me with my AI class at college. Thanks. 😊



Siddharth Agrawal · October 7, 2013 - 4:32 am · [Reply](#)→

Thanks a lot for appreciating my work. 😊



Avishek Seal · November 30, 2013 - 9:04 pm · [Reply](#)→

i didn't get here... why the heuristic value of A is 2... it should be  $1 + 2 = 3$  .. isn't it??



Siddharth Agrawal · November 30, 2013 - 9:14 pm · [Reply](#)→

Heuristics are made so that they give us an approximate cost of reaching the goal. If the heuristic value gives us the actual cost, then there is no point in searching for the shortest path, because we already know all the costs. For more detail, read my post on 'Search Heuristics'



Avishek Seal · November 30, 2013 - 9:24 pm · [Reply](#)→

So we can take any value ?? either equal or less than the actual cost

## Follow “Algorithmic Thoughts - Artificial Intelligence | Machine Learning | Neuroscience | Computer Vision”

Get every new post delivered to your Inbox.

Join 55 other followers

Sign me up

Build a website with WordPress.com

Follow

of the path???



Siddharth Agrawal · November 30, 2013 - 11:59 pm · [Reply](#)→

Yes, that's right. Trivially, even the constant '0' is a heuristic. In that case, A\* Search is the same as Uniform Cost Search.

## Follow “Algorithmic Thoughts – Artificial Intelligence | Machine Learning | Neuroscience | Computer Vision”

Get every new post delivered to your Inbox.

Join 55 other followers



[Build a website with WordPress.com](#)



Jeffrey Epstein · July 24, 2014 - 9:40 am · [Reply](#)→

A compelling blog. Well done. Just FYI, the Jeffrey Epstein VI Foundation just launched its new website on artificial intelligence covering robotics to virtual software platforms to explore cognition: <http://www.jeffreyepstein.net>



Maria · July 28, 2014 - 1:57 am · [Reply](#)→

we have to assign heuristics manually for each node?? is ther any heuristic function like something.....which do it automatically while coding..??



Siddharth Agrawal · August 1, 2014 - 10:39 am · [Reply](#)→

No, a heuristic is a mathematical function, which you include in your code for computation. For example, if you are playing Pacman, the distance to a food particle maybe a heuristic.



oola wallter · September 25, 2014 - 2:53 pm · [Reply](#)→

Thanks alot for the post, i really need to know how admissible heuristic work



Oldane Graham · December 5, 2014 - 5:51 pm · [Reply](#)→

I have been preparing for an AI exam in college and this really help a lot.



Limpho · December 7, 2014 - 2:00 am · [Reply](#)→

thanks for clarification. This example helped mi alot

## Leave a Reply

Enter your comment here...

← [Artificial Intelligence – Search Heuristics](#)

[Computer Vision – Feature Detection](#) →

Blog at WordPress.com.

The Origin Theme.

⌵

## Follow “Algorithmic Thoughts – Artificial Intelligence | Machine Learning | Neuroscience | Computer Vision”

Get every new post delivered to  
your Inbox.

Join 55 other followers

Enter your email address

Sign me up

Build a website with WordPress.com

Follow