



Show all posts ▾

by recent activity ▾

- | | | |
|---|--|----|
| | what does the dollar sign mean? | 4 |
| I have written my functions bfs, dfs, astar and ida that would write answer to output.txt. ... | | |
| | [DFS] Benchmarks | 12 |
| | IDA* Fringe Size | 11 |
| I keep getting zero grades for IDA* fringe_size / max_fringe_size values. They are actually... | | |
| | IDA* Implementation | 19 |
| I have quite a few doubts about the implementation of IDA* and it is incredibly difficult t... | | |
| | Conflict in Week 2 Project description. | 10 |
| I am somewhat puzzled by the examples in Week2 project. A stated objective is: "compar... | | |
| | Project does not run in Vocareum | 7 |
| Hi to all, i have submitted my project on Vocareum but it seems like it doesn't run there. ... | | |
| | IDA*: STUCK AT IDA* | 1 |
| I'm stuck at IDA* and would appreciate any true explanation. Say G is goal state, R is roo... | | |
| | [STAFF] A better test case for A* and IDA* | 96 |
| Hi, Would it be possible to get a more complex test case with output for A* and IDA*. It s... | | |
| | Project 1: Additional test cases | 42 |
| Hi, I think additional test cases could help with developing, debugging or verifying the pro... | | |
| | Initialize with? | 4 |
| How do I initialize frontier with initialstate in the search tree? I just started python and I h... | | |
| | Difference between search_depth and cost_of_path | 3 |
| ¿Is there any scenario when search_depth and cost_of_path have different values? If so, c... | | |

IDA* Implementation

question posted 21 days ago by [amalolan](#)

I have quite a few doubts about the implementation of IDA* and it is incredibly difficult to go search 10+ discussions on how to do so. So ill post a list of my doubts



here.

1. Should IDA* use a heapq or a stack (as in DFS)
2. Should we reset nodes_expanded each time the limit is increased?
3. Should we use an explored nodes list?
4. If we need to have an explored nodes list, should we add nodes already explored if they have lower $f(n)$ values?
5. Is it fine if the next limit is the min $f(n)$ of the nodes refused into the frontier?
Or should the increment be constant ?
6. What does the instructions mean by "For each iteration, you can handle node ordering as you would in depth-first search." Does that imply we need to use a stack?

Thank You

Related to: [Week 2 / Project](#)

This post is visible only to Default Group.

[enricogiannini](#)

20 days ago - marked as answer 20 days ago by [amalolan](#)



1) Looking at lectures, if we implement IDA* based on IDS design - actually an iterated DFS with an increasing threshold (depth level for IDS, state cost for IDA*) - I'm a bit confused how you will use an heapq in place of a stack. 2) Yes, i think so - not resetting the node leads to other interesting alghoritm (fringe search for example, where in subsequent iterations the exploring restart from previously dropped nodes for exceeding cost threshold) 3) Set is efficient, or dictionary.... any structure with a lightweight lookup. 4) Yes, you drop the node/path with higher cost and keep the light one. 5) One of the best strategy, in my opinion, is starting with the root cost and using the lower dropped cost as next cost. You are sure you'll never lose a path with an improper greater increment. 6) See 1) - it's the same point

[Show Comments \(5\)](#) ▼

[gainsley](#)

20 days ago - marked as answer 20 days ago by [amalolan](#)



I have been struggling to find a proper and consistent explanation about IDA* as well. Look at this : https://heuristicswiki.wikispaces.com/IDA*
This clearly says that it's a DFS (i.e. using a **STACK and NOT a priority Q**) where you cutoff at $g+h>\text{limit}$. I have to wrap my brain about optimality but one advantage (key points) of doing this is that it keep the fringe size small compared to A*. If you use a PriorityQ then your fringe size can also become very big.

[Show Comments \(3\)](#) ▼

[Add a Response](#)

3 other responses

HCL33

20 days ago



1. heapq, similar to A*
2. don't know
3. yes, but use set() so your 'is in' comparison is O(1)
4. same as A*
5. design decision
6. No. It means use reverse UDLR as we did for DFS.

Ok thanks a lot mate.

posted 20 days ago by [amalolan](#)

Explain why heapq?? :-/

posted 19 days ago by [AHoebeke](#)

IDA* uses a stack, not a heap. It doesn't store nodes in a priority queue. It just uses fn value to compare it to the threshold (depth boundary) and increase it to the minimum of the fn values that exceeded the previous threshold.

posted 13 days ago by [StathisPeioglou](#)

Add a comment

[JimRobertson](#)

18 days ago



Here is description from the original research paper on IDA*: "Depth-first iterative-deepening can also be combined with a best-first heuristic search such as A* [6]. The idea is that successive iterations correspond not to increasing depth of search, but rather to increasing values of the total cost of a path. For A*, this total cost is composed of the cost so far in reaching the node (g) plus the estimated cost of the path from the node to a goal state (h). Iterative-deepening-A* (IDA*) works as follows: At each iteration, perform a depth-first search, cutting off a branch when its total cost (g + h) exceeds a given threshold. This threshold starts at the estimate of the cost of the initial state, and increases for each iteration of the algorithm. At each iteration, the threshold used for the next iteration is the minimum cost of all values that exceeded the current threshold." So, at the core of this, it is a depth-first search, which seems to imply a stack for fringe. (An alternative is to use recursion, in which case it's actually the programming runtime that's providing the stack.) At each iteration of outer loop, the A* cost estimate isn't used for choosing the order of what to explore next; rather, the A* cost estimate is used for deciding whether a child branch should be pursued or cut off. So instead of limiting the depth-first search by a constant node depth at each iteration, it's the A* metric that's controlling what gets explored.

Add a comment

Support

[sonnywang](#)

14 days ago



I just found out in my notes (the lecture "search algorithms revisited" that IDA* is using A* to search the subtree, which means it should be using min-heap like what is used in A*. Not doing DFS search for every f limit.

Once again, the original IDA* research paper says "At each iteration, perform a depth-first search."



posted 13 days ago by [JimRobertson](#)

@sonnywang ,so that is why make IDA vague to understand



posted 8 days ago by [dbsx](#)



Showing all responses

Add a response:

Preview

Submit



© 2012-2017 edX Inc. All rights reserved except where noted. EdX, Open edX and the edX and Open EdX logos are registered trademarks or trademarks of edX Inc.

POWERED BY
OPENedX®

Support

