

Forum: [Hack Forums](#)

Thread: [Finding a path in an array](#)

Barinade - February 4, 2019, 11:53 pm

So, I have something like

array[0] = [1, 4, 6, 9, 22]

array[6] = [4, 10, 0]

array[4] = [6, 7, 9, 0]

array[9] = [4, 11]

I've thrown in some random numbers, but as you can see, the arrays are linked, as in one of the values in one array is the index of another array

Array 0 has values 4, and 9, both of which are array[4] array[9], etc

I need to find the shortest path between two upper levels of the array

Say I wanted a path from 9 to 0, the path would be "9, 4, 0" because 9 and 0 are not connected, but 9=>4 and 4=> 0 works

I need this to work for any path length, so if I needed to go from array 80 all the way to 1 it might take a few steps, whereas my example only takes 2 steps (9 to 4, then 4 to 0)

I've tried recursion, but my brain hurts, so now I'm asking you guys

pointandlaugh - February 5, 2019, 3:35 am

Took me a couple of read throughs to get the problem, but then I realised it's a directed graph structure. Still haven't racked up enough posts to give you links, but search for the following: Directed graphs

Dijkstra's Shortest Path Algorithm (not really an applicable solution here, since weights are equal between arrays, but great background reading)

Depth first search

Breadth first search

Those are the easiest ones you could implement for that problem. If this is an assignment and you want to know practical applications of this, games use similar patterns for AI pathfinding and action planning.

SIM-3364521 - February 5, 2019, 4:40 am

I just made a quick class solving this problem using breadth first search.

I know using two stacks is somewhat unelegant and so on.

But i made this real quick and hope it helps you:

<https://pastebin.com/m8uVaeHi>

If you got problems with it let me know :)

Barinade - February 5, 2019, 5:08 am

(Feb 5, 2019 05:40 AM)Erarnitox Wrote: [\[CITE\]](#) I just made a quick class solving this problem using breadth first search.

I know using two stacks is somewhat unelegant and so on.

But i made this real quick and hope it helps you:

<https://pastebin.com/m8uVaeHi>

If you got problems with it let me know :)

Oof, I was in the middle of doing it with Dijkstra's algorithm, because there is a measurable distance between the nodes, but it's not entirely necessary, at least not for now. Your method works just fine, I appreciate you

SIM-3364521 - February 5, 2019, 5:12 am

(Feb 5, 2019 06:08 AM)Barinade Wrote: [\[CITE\]](#) Oof, I was in the middle of doing it with Dijkstra's algorithm, because there is a measurable distance between the nodes, but it's not entirely necessary, at least not for now. Your method works just fine, I appreciate you feel free to post your solution here as well as i think mine isnt the most elegant one. dijkstra is normally my go to as well but i actually didnt know how to implement it without weighted paths. So i would actually be interested in how you did it :)

SIM-3364521 - February 5, 2019, 5:13 am

(Feb 5, 2019 06:08 AM)Barinade Wrote: [\[CITE\]](#) Oof, I was in the middle of doing it with Dijkstra's algorithm, because there is a measurable distance between the nodes, but it's not entirely necessary, at least not for now. Your method works just fine, I appreciate you feel free to post your solution here as well as i think mine isnt the most elegant one. dijkstra is normally my go to as well but i actually didnt know how to implement it without weighted paths. So i would actually be interested in how you did it :)

SIM-3364521 - February 5, 2019, 5:13 am

(Feb 5, 2019 06:08 AM)Barinade Wrote: [\[CITE\]](#) Oof, I was in the middle of doing it with Dijkstra's algorithm, because there is a measurable distance between the nodes, but it's not entirely necessary, at least not for now. Your method works just fine, I appreciate you feel free to post your solution here as well as i think mine isnt the most elegant one. dijkstra is normally my go to as well but i actually didnt know how to implement it without weighted paths. So i would actually be interested in how you did it :)

+Haze - February 5, 2019, 7:58 am

Are you by any chance trying to solve the travelling sales man problem? Or is this just a regular finding the shortest path? Because if so, you can always look into genetic algorithms or swarm intelligence as an alternative solution. It's quite an interesting topic.

lockhead - February 5, 2019, 8:49 am

(Feb 5, 2019 08:58 AM)+Haze Wrote: [\[CITE\]](#) Are you by any chance trying to solve the travelling sales man problem? Or is this just a regular finding the shortest path? Because if so, you can always look into genetic algorithms or swarm intelligence as an alternative solution. It's quite an interesting topic.

How would you use swarm intelligence to find the shortest path?

+Haze - February 5, 2019, 12:48 pm

(Feb 5, 2019 09:49 AM)lockhead Wrote: [\[CITE\]](#) How would you use swarm intelligence to find the shortest path?

The essence of swarm intelligence is that you use the collective behaviour of agents to solve a problem. One way you can use SI to find the shortest path is to use the Ant Colony Optimization algorithm. ACO is inspired by the behaviours of ants in the real world, and it's a very good algorithm for finding "good enough" solutions to the travelling salesman problem. I say good enough because

TSP is an NP-HARD problem, which means that it takes a seriously long amount of time to find the single "best" solution, that's why the Ant Colony Optimization Algorithm is good because it can help us to find a good enough answer that is time effective.

lockhead - February 7, 2019, 2:33 am

(Feb 5, 2019 01:48 PM)+Haze Wrote: [\[CITE\]](#) The essence of swarm intelligence is that you use the collective behaviour of agents to solve a problem. One way you can use SI to find the shortest path is to use the Ant Colony Optimization algorithm. ACO is inspired by the behaviours of ants in the real world, and it's a very good algorithm for finding "good enough" solutions to the travelling salesman problem. I say good enough because TSP is an NP-HARD problem, which means that it takes a seriously long amount of time to find the single "best" solution, that's why the Ant Colony Optimization Algorithm is good because it can help us to find a good enough answer that is time effective.

Interesting. I've heard of SI before but never thought it as viable substitution to path finding algos.

+Haze - February 7, 2019, 6:56 am

(Feb 7, 2019 03:33 AM)lockhead Wrote: [\[CITE\]](#) Interesting. I've heard of SI before but never thought it as viable substitution to path finding algos.

Right, yeah it's used a lot, algorithms that find "good" enough solutions are very popular, I'm sure Google use it for their map system or something.