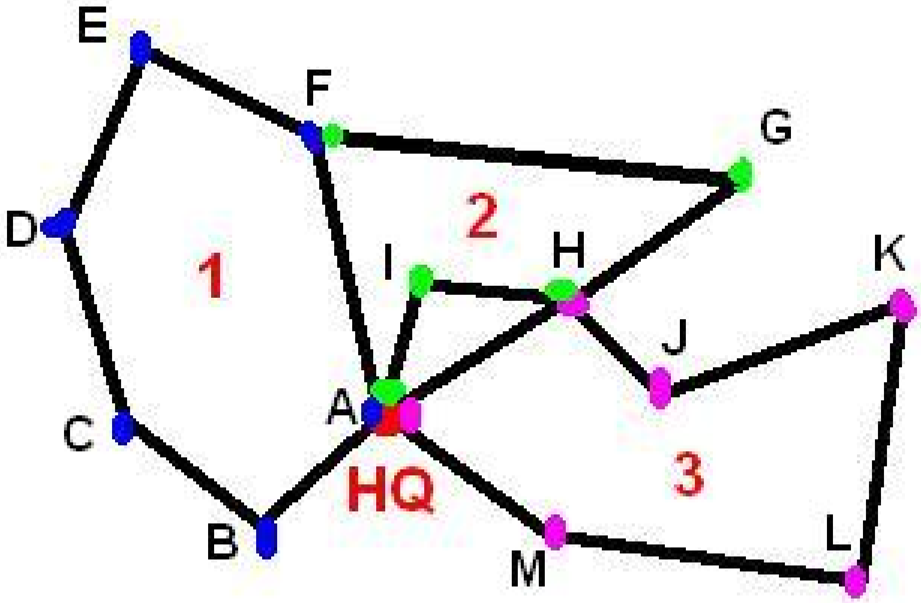
A Route Assignment Program Algorithm

Asked 12 years, 10 months ago Modified 12 years, 10 months ago

Viewed 695 times

2

What im trying to do, is create a program that will assign a route for a driving test. there will be three diffrent routes, linked together at certain points. Never should there be more than one student at a point of intersection.



Best way to solve this is to schedule these interection points by time.

This isnt my only problem, i will need routes to be equally distributed to examiners. So route 1 will be given to examiner 1 route 2 - examiner 2 route 3- examiner 3...

The Real Baumann suggested this:

> Calculate collision times from start.
>
> Route 1 has 6 points. `{A,B,C,D,E,F}`
>
> Route 2 has 5 points. `{A,F,G,H,I}`
>
> Route 3 has 6 points. `{A,H,K,L,M,N}`
>
> Possible Collisions at: `{A,F,H}`
>
> So you need to calculate the following times:
>
> Route 1: A->F, A->A
>
> Route 2: A->F, A->H, A->A
>
> Route 3: A->H, A->A
>
> From here you can calculate time differences that create a collision.
>
> If it takes you 20 minutes to go from route 1A to Route 1F and 5 minutes to get from Route 2A to Route 2F, then you know a collision will occur if start an appointment on Route 2 exactly 15 minutes after you began an appointment at Route 1.
>
> Then you would have a set of non-working collisions:
>
> Route 1 & 2 collide at: 15, 25, 40
>
> Route 1 & 3 collide at: 25, 30

> Route 2 & 3 collide at: 30, 40, 45

This i can understand to a point. But in terms of an algorithm i dont know where to start. IF someone could help me with some pseudo code to work off, or something to make it clearer in my own mind. it would help a lot.

algorithm pseudocode

Share

Improve this question

Follow

edited Feb 2, 2012 at 16:13

Security Hound

2,557 ● 3 ● 25 ● 44

asked Feb 2, 2012 at 15:35

user1081326

425 ● 5 ● 10 ● 20

1 Also, can students leave and return to HQ at the same time, or does that break your intersection rule?
– The Real Baumann Feb 1, 2012 at 18:59

1 "whats the best way about coding this" ? Well certainly not getting others think for you :) – Mehdi LAMRANI Feb 1, 2012 at 19:14

1 I'd also like to mention that all of your points that don't collide with another route don't need to be there. Just remove it entirely. in the case of route one, you only need to go from the start, to the 2nd to last point, to the end (if going clockwise). The second route only has 3 points, as does the 3rd. Technically this is just a data change, not an algorithm

change, but it does mean that you no longer need to support the non-colliding cases. – Servy Feb 1, 2012 at 19:36

1 It would be nice if you mentioned this was a followup. I figured that **The Real Baumann** was some fancy algorithm. You are making it far to complex. You know which routes CANNOT be scheduled because they could collide although that entirely depends on the speed of the drivers. The answer you recieved on your previous question gave you EVERYTHING you need to solve this problem. We except people to bring something to the Table. – Security Hound Feb 2, 2012 at 16:12

1 traffic light would be helpful. – user482594 Feb 2, 2012 at 16:53

2 Answers

Sorted by: Highest score (default)

4

You should be able to calculate collision times from start.

Route 1 has 6 points. `{A,B,C,D,E,F}`

Route 2 has 5 points. `{A,F,G,H,I}`

Route 3 has 6 points. `{A,H,K,L,M,N}`

Possible Collisions at: `{A,F,H}`

So you need to calculate the following times:

Route 1: A->F, A->A

Route 2: A->F, A->H, A->A

Route 3: A->H, A->A

From here you can calculate time differences that create a collision.

If it takes you 20 minutes to go from route 1A to Route 1F and 5 minutes to get from Route 2A to Route 2F, then you know a collision will occur if start an appointment on Route 2 exactly 15 minutes after you began an appointment at Route 1.

Then you would have a set of non-working collisions:

Route 1 & 2 collide at: 15, 25, 40

Route 1 & 3 collide at: 25, 30

Route 2 & 3 collide at: 30, 40, 45

From here you should pretty easily be able to create your schedule without collisions.

Share Improve this answer

Follow

answered Feb 1, 2012 at 19:22

The Real Baumann

1,961 ● 1 ● 14 ● 20

Thanks for taking the time to help me here, Thank you for simplifying it down in my own mind as well. So as said above, a Schedule is the best way forward here. Thanks for the reply too. – user1081326 Feb 1, 2012 at 20:02

I'm supposing you're not asking for tips on writing a super-duper algorithm that can resolve hundreds of paths

4

with thousands of intersections simultaneously. It sounds like you need something simple and serviceable, so let's aim for that.

First off, let's simplify the problem. Looking at the map, what you're really saying is something like this: If a student starts route 1 at 8am, he'll be in intersection A sometime between 8:03 and 8:05, and then in intersection B sometime between 8:07 and 8:09.

To ensure that no other students are in the intersection, you can consider Intersection A "booked" from 8:03-8:05 by the first guy, and Intersection B "booked" similarly from 8:07-8:09.

Each intersection would have its own busy/free table.

Each time you schedule a route, you book the appropriate intersections during the time you believe the student will be in them.

When looking for the earliest available time for a route, you go through the routes and consider start time X "available" for that route if the of each intersections you'd pass through on the route are available at the time you'd pass through them.

Share Improve this answer

Follow

answered Feb 1, 2012 at 19:28

CodeGnome

261 ● 1 ● 4

Thank you for taking the time to reply, i think a schedule will be the way forward on this one. – user1081326 Feb 1, 2012 at 20:03