Programming test

The Joensuu Regional Library is aiming to migrate all of Eastern Finland's municipal libraries under one system, Koha Library System. This will manifest as improved material availability for all participant libraries, as more resources for modernizing the web-library, and as license savings. This in turn will cause increased inter-library loans, higher material transportation volumes, need for more varied web-services, and generally a huge boost for participating member libraries.

Nothing good comes without a cost, however. Increased usage creates more library offenders, or outright library criminals. This creates a need to start a inter-library criminal transportation network. As a part of this effort, we need to plan our collection logistics.

Deliver, for the driver of our prisoner pickup truck driver, a tool to find a route to visit all of our libraries.

The driver needs to be able to set the time taken to travel between the three different lines, for example:
-Red line – 25 minutes
-Blue line – 15 minutes
-Green line – 5 minutes

The driver needs to be able to choose from where he starts his journey, but must always visit each library at least once. He/she can only travel between established lines.

You don't need to take into account any other specifics. Just the time to travel between libraries, because the criminals are dropped on his pickup truck on the fly.
It is enough to reach all of the libraries, i.e. the driver can find his own way back to the disposal center.

The program must output a clear, visual and traceable route plan and total traveling time.

Should you run out of time, do not hesitate to show us your plans and progress anyway.

Functional limitations:

1. The backend-program must be done in Perl.
2. Must use Asynchronous Javascript calls.

3. It must run in a Debian derivative Linux environment.
4. The program must be made available as a git repository (eg. In GitHub). Please use an obscure naming convention so that other candidates cannot find your solution.
5. Deployment:
    - To install the program, it is “git cloned” to a directory with apache2's Perl execution permissions. Zero setup time!
    - Alternatively: Any other Perl web-app with an built-in/included web server so we can test the result from our browser. We prefer Mojolicious.
6. You can still use Perl modules as we can manually install any missing Perl dependencies.
7. Business logic need not be optimal, any result will do.
8. User interface works on Chrome or Mozilla.
9. The program reads the libraries' route information from the given example text database.

Other deliverables:

1. Work logs for the time taken to complete the assignment. Short descriptions of how many hours was spent on this assignment and what was done during those hours.

Implementation notes:

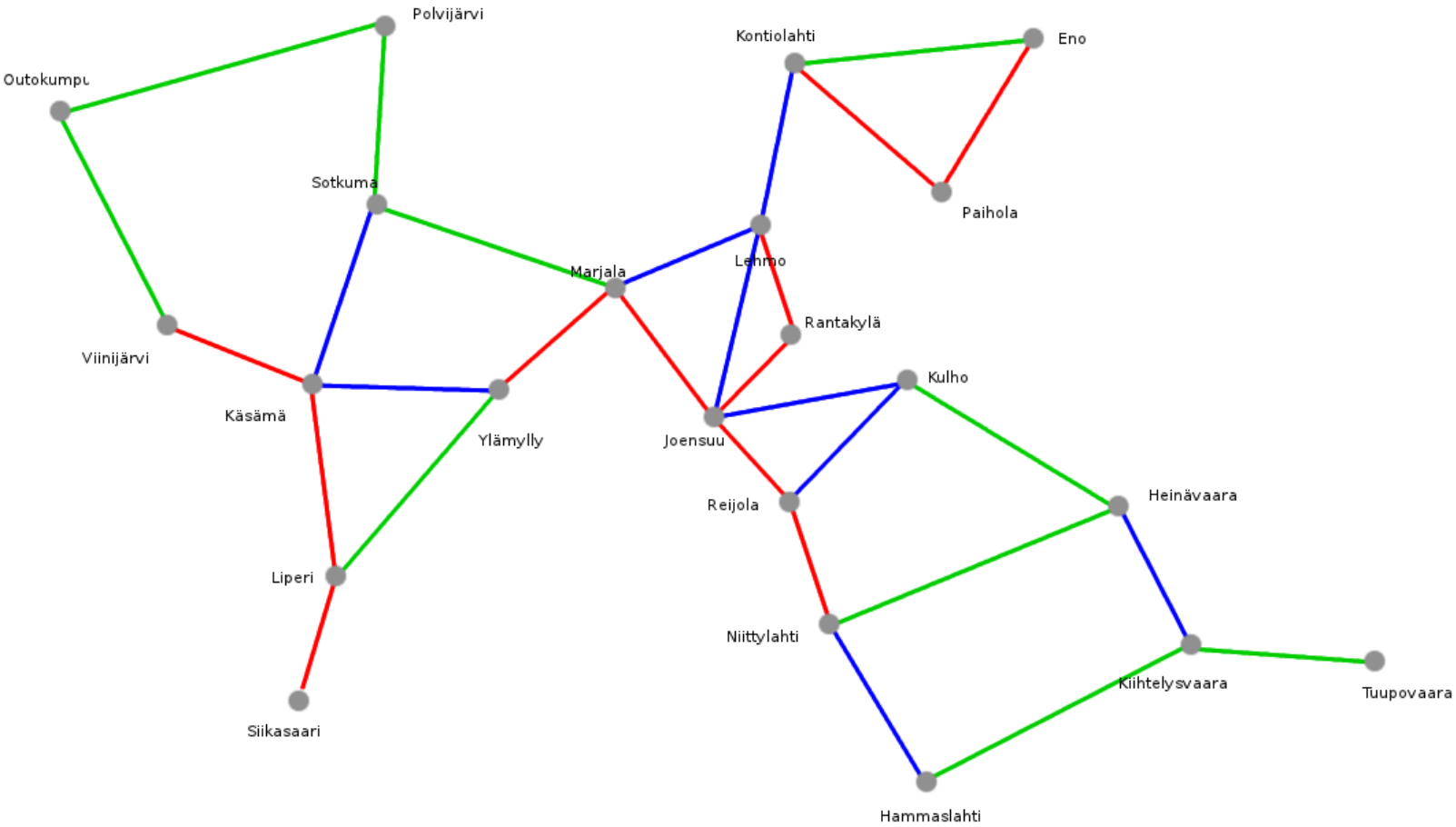
1. If accessing the git repository needs credentials, those must be submitted to the reviewer. Reviewer will not accept any invitations or create no accounts to any external service.
2. Reviewer fetches the submission with the command “git clone <repository url>” using the given credentials. Nothing more.

Here are the contents of the example text database:


```
{
  "nodes": [
    "Outokumpu",
    "Polvijärvi",
    "Sotkuma",
    "Viinijärvi",
    "Käsämä",
    "Liperi",
    "Siikasaari",
    "Ylämylly",
    "Marjala",
    "Joensuu",
    "Lehmo",
    "Rantakylä",
    "Kontiolahti",
    "Paihola",
    "Eno",
    "Reijola",
    "Kulho",
    "Niittylahti",
    "Heinävaara",
    "Hammaslahti",
    "Kiihtelysvaara",
    "Tuupovaara"
  ],
  "edges": [
    {
      "start": "Outokumpu",
      "end": "Polvijärvi",
      "color": "green"
    },
{
      "start": "Outokumpu",
      "end": "Viinijärvi",
      "color": "green"
    },
{
      "start": "Polvijärvi",
      "end": "Sotkuma",
      "color": "green"
    },
    {
      "start": "Sotkuma",
      "end": "Marjala",
      "color": "green"
    },
    {
      "start": "Liperi",
      "end": "Ylämylly",
      "color": "green"
    },
    {
      "start": "Kontiolahti",
      "end": "Eno",
      "color": "green"
    },
    {
      "start": "Kulho",
      "end": "Heinävaara",
```

```
   "color": "green"
},
{
   "start": "Niittylahti",
   "end": "Heinävaara",
   "color": "green"
},
{
   "start": "Hammaslahti",
   "end": "Kiihtelysvaara",
   "color": "green"
},
{
   "start": "Kiihtelysvaara",
   "end": "Tuupovaara",
   "color": "green"
},
{
   "start": "Sotkuma",
   "end": "Käsämä",
   "color": "blue"
},
{
   "start": "Käsämä",
   "end": "Ylämylly",
   "color": "blue"
},
      {
   "start": "Marjala",
   "end": "Lehmo",
   "color": "blue"
},
      {
   "start": "Joensuu",
   "end": "Lehmo",
   "color": "blue"
},
      {
   "start": "Lehmo",
   "end": "Kontiolahti",
   "color": "blue"
},
      {
   "start": "Joensuu",
   "end": "Kulho",
   "color": "blue"
},
      {
   "start": "Reijola",
   "end": "Kulho",
   "color": "blue"
},
      {
   "start": "Niittylahti",
   "end": "Hammaslahti",
   "color": "blue"
},
      {
   "start": "Heinävaara",
   "end": "Kiihtelysvaara",
   "color": "blue"
```

```
    },
    {
      "start": "Viinijärvi",
      "end": "Käsämä",
      "color": "red"
    },
        {
      "start": "Käsämä",
      "end": "Liperi",
      "color": "red"
    },
        {
      "start": "Liperi",
      "end": "Siikasaari",
      "color": "red"
    },
        {
      "start": "Ylämylly",
      "end": "Marjala",
      "color": "red"
    },
        {
      "start": "Marjala",
      "end": "Joensuu",
      "color": "red"
    },
        {
      "start": "Joensuu",
      "end": "Rantakylä",
      "color": "red"
    },
        {
      "start": "Rantakylä",
      "end": "Lehmo",
      "color": "red"
    },
        {
      "start": "Joensuu",
      "end": "Reijola",
      "color": "red"
    },
        {
      "start": "Reijola",
      "end": "Niittylahti",
      "color": "red"
    },
        {
      "start": "Kontiolahti",
      "end": "Paihola",
      "color": "red"
    },
        {
      "start": "Paihola",
      "end": "Eno",
      "color": "red"
    }
  ]
}
```



This is a picture representation of the library topography.