The Travel Challenge

Programming test for applicants at Satalia - 2019

You are a travel fanatic and have just managed to find three spare weeks for a holiday in your busy life! You decide to take this opportunity to fulfill a lifelong dream: visiting the most amazing places in the world.

Your location is known as LONG/LAT and you have a self-driving electric helicopter that travels on average at 80km/h. Since the helicopter is equipped with ultra-efficient solar panels, it doesn't need even need to stop for recharging. You decide to restrict yourself to visiting only UNESCO World Heritage places, of which there are three types: *cultura*l places, such as historic city centres, *natural* places, such as wildlife sanctuaries, and *mixed* places that represent both.

The challenge is to plan a travel itinerary which would let you see as many different sites as possible, starting from a given location LONG/LAT and getting back to your starting location. Your time budget is three weeks, and you assume that you spend 6 hours per site, covering for sightseeing, food/drink and sleep (which you can also do in the helicopter, since it contains a secret chest of snacks and a comfy futon). Your itinerary must have an equal number of cultural and natural world heritage sites (where mixed sites count for both natural or cultural sites).

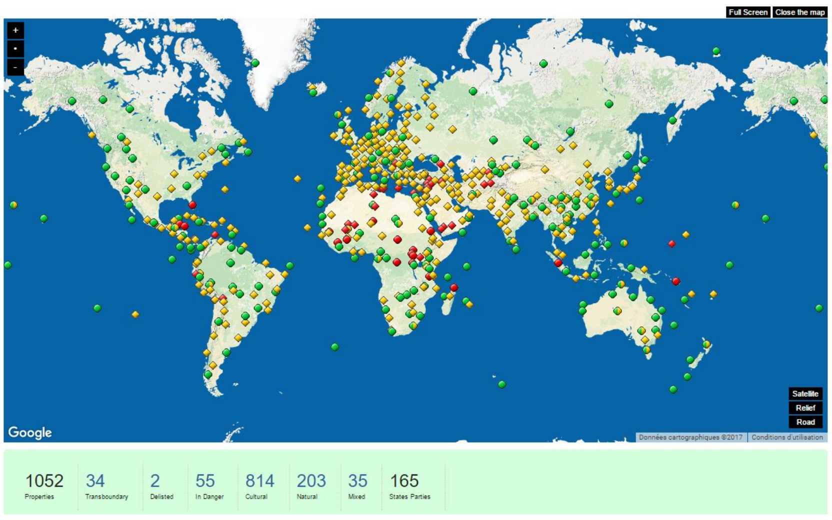
You evaluate the quality of your travel itinerary in the following way:

- Each visited world heritage *site* counts for 1 point
- Each *country* you visit counts for 2 points
- Each site that is listed as *endangered* counts for 3 points

You only receive points for the first time you visit the site/country. Your goal is to find an itinerary with a high score and that respect the requirements above.

Travel location data

There are over 1,000 UNESCO World Heritage sites in the world, which are shown in the picture below. You can find their description, including their geographic coordinates, in an Excel file which can be downloaded from http://whc.unesco.org/en/list/xls/?2018.
The data can be preprocessed in any way you like. You can convert it to a CSV file, it may be stored into a RDBMS, a graph database or into files, whichever you see fit to complete this task.



Requirements

The program must accept at least the LAT and LONG coordinates as arguments, which represent the starting and end point of your journey. The output of the program should show the travel itinerary, including the name of each site, its type (cultural or natural or combined) and its country. If you want, you can also display the travel time between each destination. Map visualisation is not required, but might come in handy for debugging. The program must accept any coordinates around the globe.
Please use Java 8 as programming language, since this is our main programming language. You can choose any database and software libraries you want (if you decide to use any), as long as you provide them (e.g. their jar-files) in your submission. Add a README file to your submission where you describe how to run your program.

Describe your approach briefly

When handing out your code, include a short paragraph highlighting your approach to solving the problem. We are particularly interested in:

- How you designed your solution
- What optimisation technique have you used (if applicable)
- What kind of coding best practices you have applied
- What kind of data structures you have used
- What considerations you had regarding computational complexity

Evaluation criteria

- How closely your program fulfills the task and how correct the proposed solutions are.
- Code quality: readability, architecture and complexity. This is a major component of this test. Your code must be as close as possible to production-level code, as if you were implementing it within an industry project.
- Your program should contain unit tests.

- How you solved the problem of selecting travel destinations. Note that this test does not focus on optimisation. A simple, well-designed approach will be worth more than a complex, poorly-implemented one. Do not favour complexity over good design, or you will be penalised for it. You are free to use optimisation libraries or external solvers, as long as you ship them with your code (as well as the licenses, if applicable).
- If you have experience in optimisation and want to implement more advanced algorithms, feel free to do so. The quality of the solutions proposed by your code will only represent a minor evaluation criterion.

In order to simplify your task

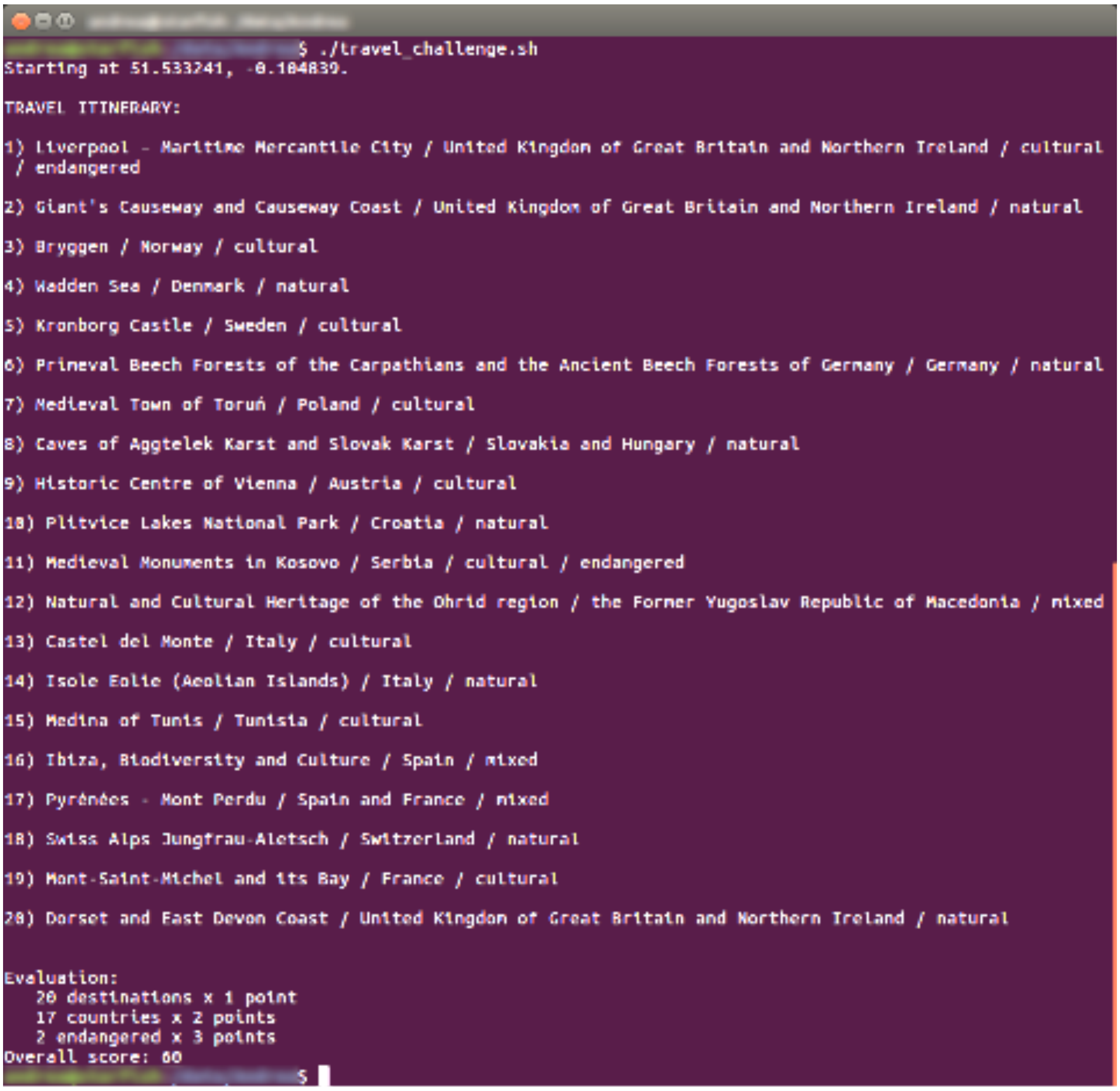
- To calculate the distance from point to point and search for the next best location, you may use a travel matrix of distances and/or travel times.
- To calculate the distance between geo coordinates you may use Haversine formula: https://en.wikipedia.org/wiki/Haversine_formula. Google it, it should be implemented in your favourite programming language.

Hints

- First try to find a simple itinerary, then try to optimise it.
- In order to test your program you could visualise the route (for instance into a map) to see how effective your algorithm is.

Example

As an example, assume you start at Satalia's head office at 40 Islington High St, London UK, N1 8XB, which has the coordinates: 51.533241, -0.104839. Below is a sample solution that visits 20 destinations. For each destination, the country and type are displayed. The overall score is displayed at the very end.



```
[redacted]$ ./travel_challenge.sh
Starting at 51.533241, -0.104839.

TRAVEL ITINERARY:

1) Liverpool – Maritime Mercantile City / United Kingdom of Great Britain and Northern Ireland / cultural
 / endangered

2) Giant's Causeway and Causeway Coast / United Kingdom of Great Britain and Northern Ireland / natural

3) Bryggen / Norway / cultural

4) Wadden Sea / Denmark / natural

5) Kronborg Castle / Sweden / cultural

6) Primeval Beech Forests of the Carpathians and the Ancient Beech Forests of Germany / Germany / natural

7) Medieval Town of Toruń / Poland / cultural

8) Caves of Aggtelek Karst and Slovak Karst / Slovakia and Hungary / natural

9) Historic Centre of Vienna / Austria / cultural

10) Plitvice Lakes National Park / Croatia / natural

11) Medieval Monuments in Kosovo / Serbia / cultural / endangered

12) Natural and Cultural Heritage of the Ohrid region / the Former Yugoslav Republic of Macedonia / mixed

13) Castel del Monte / Italy / cultural

14) Isole Eolie (Aeolian Islands) / Italy / natural

15) Medina of Tunis / Tunisia / cultural

16) Ibiza, Biodiversity and Culture / Spain / mixed

17) Pyrénées - Mont Perdu / Spain and France / mixed

18) Swiss Alps Jungfrau-Aletsch / Switzerland / natural

19) Mont-Saint-Michel and its Bay / France / cultural

20) Dorset and East Devon Coast / United Kingdom of Great Britain and Northern Ireland / natural


Evaluation:
   20 destinations x 1 point
   17 countries x 2 points
   2 endangered x 3 points
Overall score: 60
[redacted]$
```