♫ Alarm bells ring, are you listening?
Santa's sleigh has gone missing ♫

Kickoff Santa Claus Challenge

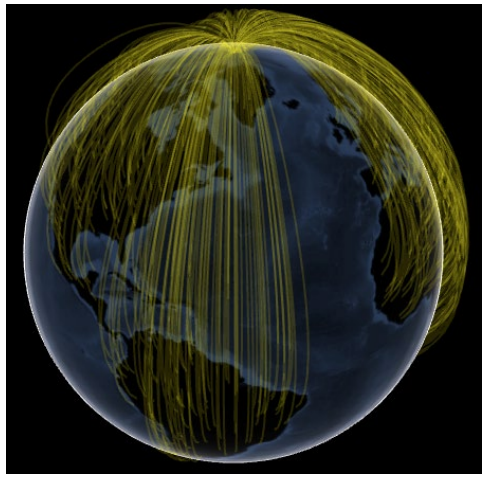
- Presentation of Challenge (10min)
- Examine task individually (15min)
- Brainstorming / Open Questions (10min)
- Form groups of 2-5 people (5min)
- Breakout sessions within the groups (20min)

The North Pole is in an uproar over news that Santa's magic sleigh has been stolen. Able to carry all the world's presents in one trip, it was considered crucial to successfully delivering holiday goodies across the globe in one night.

Unwilling to cancel Christmas, Santa is determined to deliver toys to all the good girls and boys using his day-to-day, magic-less sleigh. With so little time to pull off this plan, Santa is once again counting on Kagglers to help.

Given the sleigh's antiquated, weight-limited specifications, your challenge is to optimize the routes and loads Santa will take to and from the North Pole. And don't forget about Dasher, Dancer, Prancer, and Vixen; Santa is adamant that the best solutions will minimize the toll of this hectic night on his reindeer friends.

Your Task: Help Santa!



Your goal is to minimize total weighted reindeer weariness:

- Weighted Reindeer Weariness = (distance traveled) * (weights carried for that segment)
- All sleigh trips start from north pole (Lat=90, Long=0), then head to each gift destination in the order that a user gives, and then head back to north pole
- Sleigh has a base weight = 10
- The sleigh has a cargo weight limit = 1000 (excluding the sleigh base weight)

Mathematical Formulation

Mathematically, weighted reindeer weariness is calculated by:

$$WRW = \sum_{j=1}^{m} \sum_{i=1}^{n_j} \left[\left(\sum_{k=1}^{n_j} w_{kj} - \sum_{k=1}^{i-1} w_{kj} \right) \cdot Dist(Loc_i, Loc_{i-1}) \right]_j$$

where m is the number of trips, n_j is the number of gifts for each trip j, w_{ij} is the weight of the i-th gift at trip j, $Dist()$ is calculated with Haversine Distance between two locations, and Loc_i is the location of gift i. Loc_0 and Loc_n are the North Pole, and w_{nj}, a.k.a. the last leg of each trip, is always the base weight of the sleigh.

For example, if you have 2 gifts A, B to deliver in a trip, then the WRW is calculated as:

Dist(North pole → A) * *(base_weight + weight(A) + weight(B)) +*
Dist(A → B) * *(base_weight + weight(B)) +*
Dist(B → North pole) * *(base_weight)*

Haversine Distance

- The **haversine formula** is an equation important in navigation, giving great-circle distances between two points on a sphere from their longitudes and latitudes.
- The **great-circle** or **orthodromic distance** is the shortest distance between two points on the surface of a sphere, measured along the surface of the sphere (as opposed to a straight line through the sphere's interior). The distance between two points in Euclidean space is the length of a straight line between them, but on the sphere there are no straight lines. In non-Euclidean geometry, straight lines are replaced with geodesics. Geodesics on the sphere are the *great circles* (circles on the sphere whose centers coincide with the center of the sphere).

- φ_1, φ_2: latitude of point 1 and latitude of point 2, in radians
- λ_1, λ_2: longitude of point 1 and longitude of point 2, in radians

$$d = 2r \arcsin\left(\sqrt{\sin^2\left(\frac{\varphi_2 - \varphi_1}{2}\right) + \cos(\varphi_1)\cos(\varphi_2)\sin^2\left(\frac{\lambda_2 - \lambda_1}{2}\right)}\right)$$

Competition

- Data:

https://www.kaggle.com/c/santas-stolen-sleigh

(you can also find the data files on Moodle)

- Submission:

Please submit your solution (after registration) at

https://www.kaggle.com/c/santas-stolen-sleigh/submit

and then enter your score at

https://docs.google.com/forms/d/e/1FAIpQLSdsHGTbE5Mdu1chxiB2417i2QahYYLVmvIyvvvBOKASC3Pmzw/viewform?usp=sf_link

- Deadline: **Monday, 20.12.2021, 23:59**