Planet

Andrei Giurgiu

March 16, 2012

1 Easy

Since Heidi can use the time machine for free, she can travel at no cost between points lying at the same latitude. The shortest walking path between the starting point and a city, on latitudes ϕ_S and ϕ (in degrees), respectively, measures $|\phi_S - \phi| \cdot \pi R/180$ furlongs. The solution is just to count for how many cities this quantity is less or equal than Heidi's walking budget of D furlongs.

2 Medium

The easy solution to this problem is to notice that the graph that results when one draws the border is planar. The faces of this graph correspond to countries, edges to borders and vertices to border tripoints. There are no graph vertices with degree greater than 3 since that would imply that the (at least 4) adjacent faces (countries) would have capitals lying on the same circle, a case which was excluded from the start. Euler's formula for planar graphs (embedded on a sphere) tells us that $f - e + v = 2$, where f, e and v are the numbers of faces, edges, and vertices, respectively. We know that f is just the number of cities n, and since each vertex is adjacent to exactly 3 edges and each edge to 2 vertices, we get the additional relation $2e = 3v$. Then we obtain a very simple formula for the answer, namely $e = 3(n-2)$ (for $n > 1$).

There exists a much harder solution to the problem, which involves computing the positions of all the borders. Such a solution running in $O(n^2 \log n)$ time would also be accepted.

3 Hard

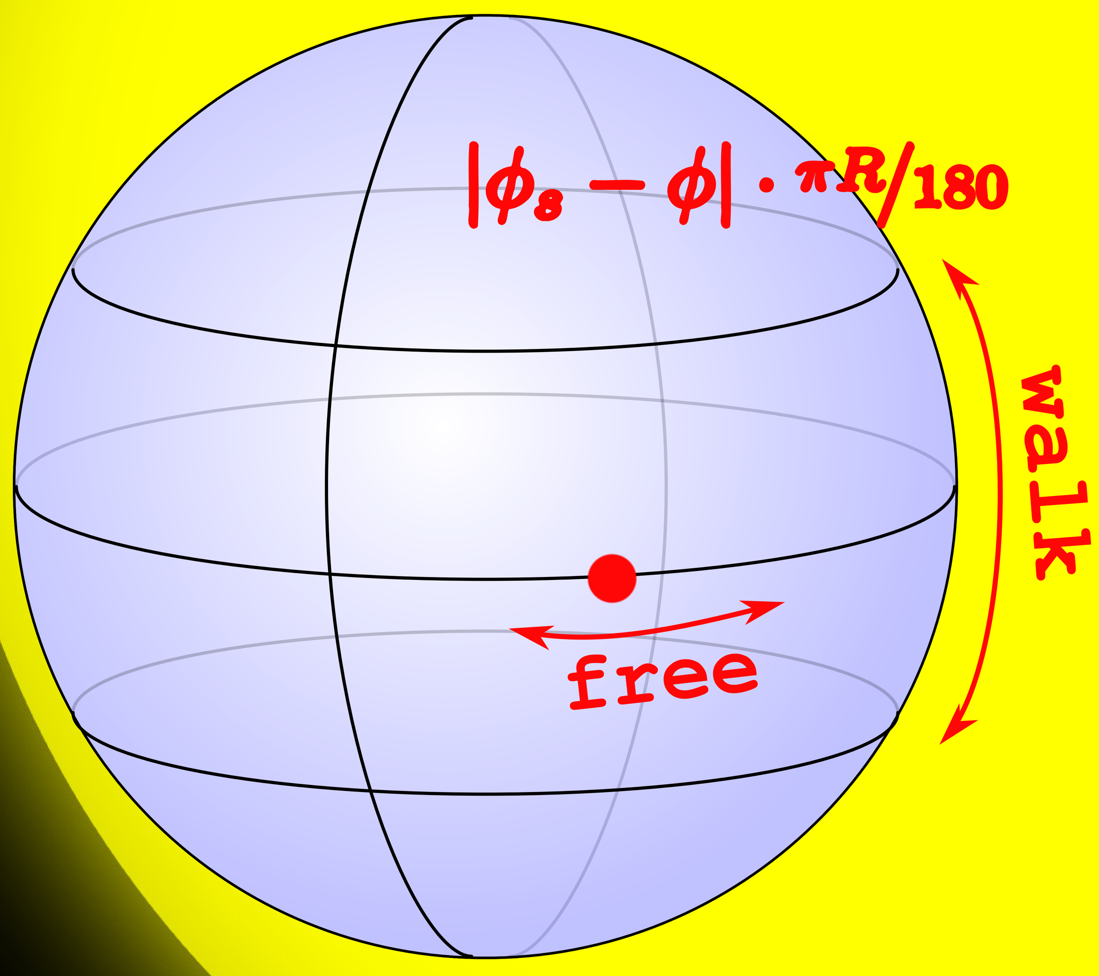
Fix two cities, A and B. Let $I(A;B)$ be the set of points on the equator that are closer to A than to B. Note that the whole equator is just the union of $I(A;B)$ and $I(B;A)$ and the two points of equal distance to A and B, call them C_1 and C_2. Suppose first that A and B were the only cities on the planet. Then the boundary between their countries would be a big circle (purple in the figure), which intersects the equator at two opposite points, exactly C_1 and C_2. This means that $I(A;B)$ and $I(B;A)$ are two intervals of equal length (colored green and cyan in the figure). Note that the vector $A - B$ (colored with red in the figure) is perpendicular to the boundary circle of A and B. To compute the positions of points C_1 and C_2, we project the vector $A - B$ onto the xOy plane, and then measure the angle phi (the azimuth, or longitude of this vector). Fortunately, there is the function $\text{atan2}(y, x)$ that does exactly this, so $\varphi = \text{atan2}(a_y - b_y, a_x - b_x)$ (in radians). The longitudes of points C_1 and C_2 are then $\varphi - \pi/2$ and $\varphi + \pi/2$. With this, we have completely characterized the quantity $I(A;B)$.

We are now interested in the nonempty sets of the form $\cap_{P_i \neq A} I(A;P_i)$, where $\{P_i\}$ is the list of cities. For each A we determine this quantity by iteratively intersecting the intervals $I(A;P_i)$. The total complexity of this method is $O(n^2)$.

The last issue is the one dealing with converting spherical coordinates (longitude ϕ and latitude θ) into cartesian ones. We use the formulas $x = cos(\phi)cos(\theta)$, $y = sin(\phi)cos(\theta)$, $z = sin(\theta)$, after transforming ϕ and θ into radians.

HC2 2012
Planet

Heidi has to find her way around the Planet at the Center of the Universe. She can freely move along the latitude rings, but has to walk along longitude rings.



(1) Easy subtask:

Count the number of Cities where the walking distance

$$|\phi_s - \phi| \cdot \pi R/180$$

is less than D.

(2) Medium subtask:

* The country map is a planar graph. Each face corresponds to a town, and country borders are the edges.
* No vertex has degree > 3 (because no 4 towns lie on same circle)
* We can use Euler's formula relating faces (f), edges (e) and vertices (v):

$$f - e + v = 2$$

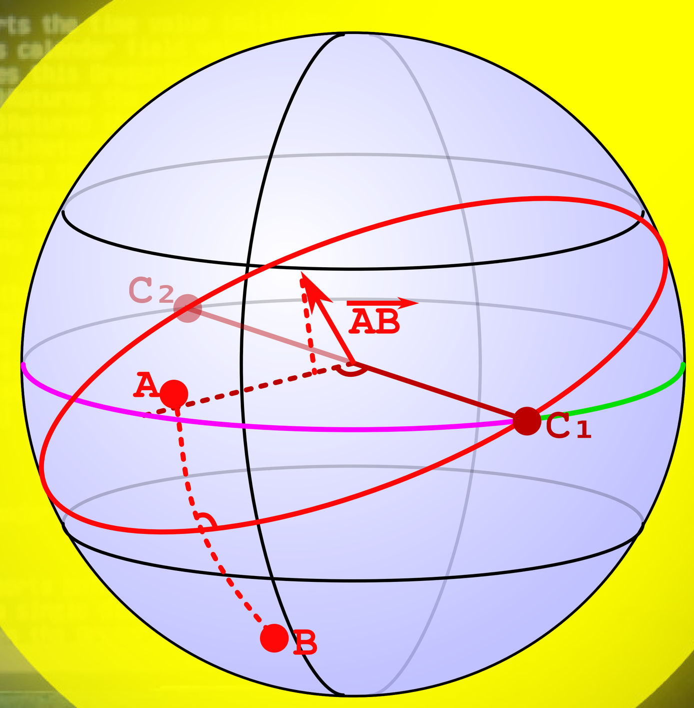
* Each vertex has 3 edges, each edge has two vertices

$$2e = 3v$$

=> Answer: e = 3*(f-2)

(3) Hard subtask:

* For every pair of cities (A,B), find which interval of the equator is closer to A than to B. See figure below. Call this I(A,B).
* Find the regions that are closer to A than for all B ≠ A: iteratively intersect the intervals I(A,·)
* Count for how many A this intersection is non-empty.



PolyProg

EPFL
ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

open systems

AdNovum