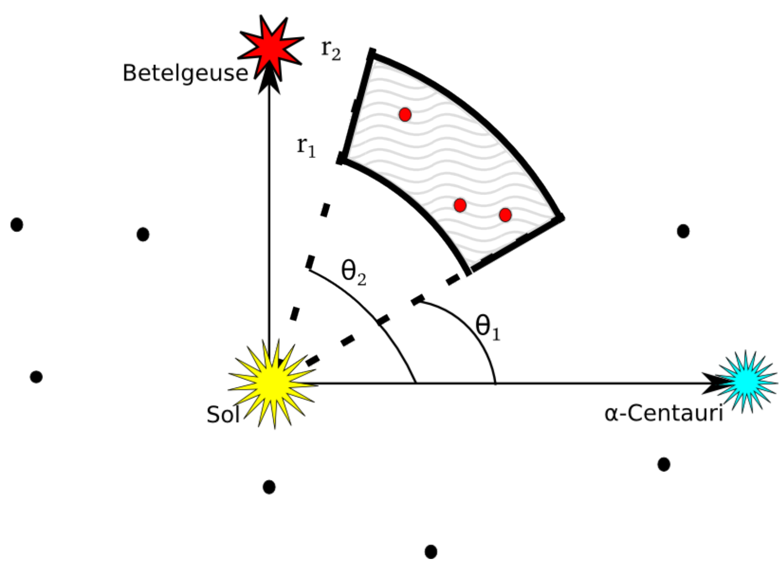
CS558: Homework 2
Due date: Thursday, Feb. 24 2011

1 Summary

It is the year 40XX and you are developing an asteroid detection system. Your main task is to locate all of the asteroids within a trimmed annular region in the solar system. Over the timescales that we are concerned with, the position of the asteroids and bodies in our system will not change much, and so we may regard it as static. Also, since the asteroids are very small relative to planets and space ships, we make the simplifying assumption that each of them can be represented by a single point. Moreover, all of the bodies we shall be interested in do not deviate much from the elliptic plane, and so we can regard them as two-dimensional quantities. Finally, to give a set of coordinates for our points, we will use the distant stars α-Centauri and Betelgeuse (which we shall assume to be perpendicular to one another relative to the sun) to define a set of axes with our sun (Sol) at the origin. The situation can be visualized as follows:



The input to your program will come in two parts, first there will be some telemetry data which will describe the position of each asteroid, and second there will be a sequence of range queries. The telemetry data consists of a sequence of names and positions for each asteroid. The position for each asteroid is given with respect to its signed distance along the axis pointing towards α-Centauri and Betelgeuse respectively, which we shall simply abbreviate by a pair of coordinates, (x, y).

The second part of the input will be a collection of range queries, each of which specifies a trimmed annular region in the elliptic plane (as shown in the figure). This range will be specified by a tuple of 4 numbers $\theta_1, \theta_2, r_1, r_2$, where θ_1, θ_2 are angles measured relative to the α-Centauri (aka x) axis, oriented toward the Betelgeuse (aka y) axis; and the parameters r_1, r_2 determine the lower/upper bounds on the annular region. This region can be described in terms of the above coordinate system as follows:

$$R(\theta_1, \theta_2, r_1, r_2) = \{(r\cos(\theta), r\sin(\theta)) : \theta \in [\theta_1, \theta_2], r \in [r_1, r_2]\} \quad (1)$$

For each of these regions, you must find all of the asteroids which are contained within the range.

1.1 Input Format

The input to your program will come from stdin. The input will begin with a pair of integers, $1 \leq n, m \leq 10^8$ representing the number of asteroids and range queries respectively. This will then be followed by n lines describing the state of all the asteroids in the solar system. Each of these line will begin with 1 to 16 non-whitespace characters representing the name of the asteroid, followed by a pair of whitespace separated double precision numbers $-100 \leq x, y \leq 100$ representing the distance along the α-Centauri and Betelgeuse axes respectively in astronomical units. After this, there will be another m lines describing a collection of range queries. Each range query will consist of four double precision numbers, respectively $\theta_1, \theta_2, r_1, r_2$ as in Eqn. 1 which will again be separated by whitespace. The first two numbers, $0 \leq \theta_1 \leq \theta_2 \leq 1080$, represent the angles in Eqn.1 given in degrees. The second two numbers, $0 \leq r_1 \leq r_2 \leq 200$ represent the inner and outer radii of the range, again given in astronomical units.

1.2 Output Format

Your program should write to stdout.[1] For each of the m range queries, you should sequentially output a line which says "`Case` i:", where i is the number of the range query starting from 1. This will then be followed by a lexicographically[2] ordered list of all asteroids contained in region within this region. If there are no asteroids within the region, then you should instead output the string "`No asteroids in range`". The output for each range query should be followed by a blank line.

1.3 Example Input

```
3 2
2009DD45        0        10
Apophis        10       1.5
Pluto        -100      -100
  0 180  2 20
360 720 30 40
```

1.4 Example Output

```
Case 1: 2009DD45 Apophis

Case 2: No asteroids in range
```

2 Written Assignment

For the written assignment, you should answer the following three questions:

1. Describe what your program does and why it correctly solves the problem.
2. What is the time and space complexity of your algorithm as a function of m, n and k, where k is the sum total of the number of asteroids contained in each range?

[1] You may also choose to write to stderr for debugging purposes, but this output will not be graded.
[2] Use the ascii codes to order individual characters.

3. Supose that instead of locating all of the asteroids in a region, instead you want to find the closest asteroid to a given point. Give a high level algorithm to do this using your data structure, and what would be it expected time complexity of this operation. Is it possible to do better, and if so, using which efficient data structure?

3 Grading

You will receive 50% credit for a correct program, and another 50% for a correct analysis and answer to each of the above questions.

4 Extra Credit

You may earn an additional 5% extra credit if you implement and describe a solution which solves this problem in $O(n \log^2(n) + m \log(n) + k \log(k))$ time and $O(n \log(n))$ space, where k is the total number of asteroids reported by all range queries.

5 What to turn in

You should should turn in a gzipped tarball named "*yourname*`.hw2.tar.gz`", where *yourname* is your last name. This archive should contain the following three things:

1. Your source code.
2. A Makefile.
3. A README which contains your answer to the written portion of the assignment.

Email your solution to mikolalysenko@gmail.com with the subject "CS 558 HW2".