

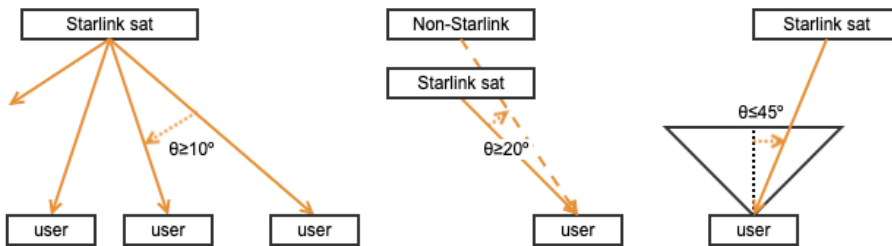
Starlink Beam Planning Tech Test

To provide internet to a user, a satellite needs to form a "beam" towards that user, and the user needs to form a beam towards the satellite. If they are both pointing at each other, they can form a high-bandwidth wireless link.

The Starlink satellites are designed to be very flexible. For this problem, each satellite is capable of making up to 32 independent beams simultaneously. Therefore, one satellite can serve up to 32 users. Each beam is assigned one of 4 "colors" (which correspond to the particular frequency used to serve that user), which is necessary to allow a single satellite to serve users that are close to one another without causing interference.

There are a few constraints which limit how those beams can be used:

- On each Starlink satellite, no two beams of the same color may be pointed within 10 degrees of each other, or they will interfere with each other.
- Other non-Starlink satellites (interferers) might be trying to provide service in the same location we are. To protect them, beams from our satellites must not be within 20 degrees of a beam from any non-Starlink satellite (from the user's perspective).
 - For simplicity, we assume that every non-Starlink satellite is always providing service to all users all the time.
- From the user's perspective, the beam serving them must be within 45 degrees of vertical. Assume a spherical earth, so all user normals pass through the center of the earth (0,0,0).



Problem:

Given a list of users, Starlink satellites, and non-Starlink satellites, figure out how to place the beams to serve the most users, respecting the constraints above. It is most important to not violate any constraints, it may not be possible to cover all users in some of the provided input files.

The inputs to your program will come from a text file. Your program will receive the name of that text file on the command-line as a single argument. The input file will look like the following:

```
# Lines that start with '#' are considered comments, and should be ignored
#
# The input is the id and position of the Starlink satellites, users, and interferers:
#   sat <id> <position x y z>
#   user <id> <position x y z>
#   interferer <id> <position x y z>
#
# Positions are in kilometers from the center of the planet.

# User on the equator at the Prime Meridian, satellite 550km overhead
user 1 6371 0 0
sat 1 6921 0 0

# Two users close to the North Pole, satellite 550km overhead
user 2 0 0 6371
user 3 111.189281412 0 6370.02966584
sat 2 0 0 6921

# Interferer satellite in GEO at 180 degrees West (opposite the Prime Meridian)
interferer 1 -42164 0 0
```

Note that the provided positions use the [earth-centered, earth-fixed \(ECEF\) coordinate system](#) where the x-axis goes from the center of the earth through the point where the prime-meridian and equator intersect, the z-axis goes from the center of the earth through the north pole, and the y-axis completes the right hand coordinate system.

The output from your program should go to standard out, and should describe the allocation of beams between each satellite and its users.

```
sat 1 beam 1 user 1 color A
sat 2 beam 1 user 2 color A
sat 2 beam 2 user 3 color B
```

Passing your output through our validation script will double-check that you haven't violated any constraints, and will tell you about the user coverage of your output. The evaluate script uses python3.7, so keep in mind that you will need to have that installed in order to properly run it. Our validation script will accept comments in your output, this is a useful property if you want to debug the output of your solver.

```
$ ./my_executable test_cases/00_example.txt |./evaluate.py test_cases/00_example.txt
Checking user coverage...
45.52734375% of 10240 total users covered.
Checking each user can see their assigned satellite...
    All users' assigned satellites are visible.
Checking no sat interferes with itself...
    No satellite self-interferes.
Checking no sat interferes with a non-Starlink satellite...
    No satellite interferes with a non-Starlink satellite.
Solution passed all checks!
```

Success Criteria:

Valid solutions must not violate constraints. Valid solutions shall be judged on the percentage of users that are covered. *Note that the simplest possible solution might be suboptimal.* Many of the testcases are impossible to get to 100% user coverage on, so you should try to gain confidence that your solution is the best that can be attained.

Your solution has to be able to generate results in a reasonable amount of time (shouldn't allocate more than 1 GB of memory or take more than 15 minutes for any individual test on a standard computer).