# Analysis of task minecraft

Solution, idea: Dimitar Shapatov; Statement, analysis: Kiril Zashev

## First stage

Let us look at two ways to solve the first stage:

- Divide and conquer: let us say that at the moment we want to find the positions of all portals in the interval $[l, r]$. We split the interval into two equal parts, and run divide and conquer for the two halves. If some interval has a number of portals equal to its length, we directly take the positions of the portals and stop.
- Binary search: the number of portals for every prefix, by increasing length, is monotone. This way we can find the portals using only queries with $l = 1$. This approach on average uses twice as many queries and gives about 93 points.

## Second stage

If you have not understood what is being asked – we have starting vertices (the portals in this case) and we want to find the shortest distance from some starting vertex to some other end city (the Nether Fortresses) in a graph.

Let us look at two possible solutions:

- Slow solution: we run Dijkstra from every starting vertex and take its minimum distance to some end city.
- Fast solution: we push all starting vertices into the priority queue before we start Dijkstra. The initial distance to every starting vertex is 0, and all starting vertices are processed simultaneously, and this is an argument with which we can prove the correctness of this Dijkstra.

## Third stage

Let us look at two possible solutions:

- Slow solution: we go over every pair of coordinates until we hit NONE.
- Fast solution: we do a two-dimensional binary search. We will have to maintain two pointers each for the interval of the search space – one for the first dimension and another for the second dimension.