Tarzan of the Apes

Problem category: Graph Theory
Expected difficulty: 1800

Solution

As you might have guessed, the most straightforward solution to this problem is Dijkstra's algorithm. Indeed, we can construct a digraph in which the edges indicate that a source tree can reach a target tree if the Euclidean distance between them is not greater than the height of the source tree. Then, we can apply the usual Dijkstra's algorithm, which runs in $O(n^2)$ for dense graphs.

We just have to be mindful of a few things:

- The memory constraint prevents us from allocating enough memory to maintain the whole graph, which amounts to $O(n^2)$ to hold all the edges. This can be easily circumvented.
- We must use the actual Euclidean distance (which implies floating-point square root computation). Otherwise, the algorithm would yield an incorrect result.

To improve the runtime, we can do the following:

- Sort the trees by their x-coordinate. This will allow us to iterate over neighbouring trees in the x-axis, until the horizontal distance exceeds the height of the source tree.
- Skip a target tree that has been reached with a traveled distance not greater than that of the source tree (since it won't get a better score). This will save the cost of a square root computation.

Complexity

Since we need to run Dijkstra's algorithm on pairs of trees, the time complexity is either $O(n^2)$ or $O(n^2 \log n)$, depending on implementation.

On the other hand, since we are not building the actual graph, the overall memory complexity is $O(n)$.