# Interpolation

## Original Idea

I deduced that the rules for interpolation must be the following:

* Every point in pts to get elevation for must be within two indexes of another pt to get elevation for
* It is preferable to have distance of 2 whenever possible; minimize 1s
* Only points that have included references may have elevation data gathered for them
* All pts must either have direct elevation data, or be within 1 index of two other pts with direct elevation data.

Therefore, an algorithm must include the following:

* A table that has values for the status of each point
* The start, end, and breaking points must have direct elevation data
* The table should be updated with references constantly (i.e. ANY added interpolation point must be spread out, since it is duplicate data)

After attempting to build a solution, I deemed that this would be too complicated, in light of the idea I had of using the connections on intersections.

## New Idea

I had the idea of using the intersections data, since it is already broken up at this point into what I do and don’t need, and it does not have the problem of updating intersections.

There will be an array that will hold key pairs of connections. The reverses will not be permitted to exist. The necessary pairs will be taken from the list of pre-generated routes, so that unneeded connections not used by two used intersections will not be counted. I had not accounted for that in the previous algorithm, because I just thought of that.

I implemented this algorithm the night of 6/8/15, and tested it on 6/9/15, but an error with my program caused an OVER\_QUERY\_LIMIT error (daily limit reached: confirmed) because the queries were unacceptable, but still registered. On 6/10/15, I fixed this problem by making each query 300 pts long, instead of that while loop.

This approach works. I ran two tests. The first was my visual inspection of the first forty points: all of them made sense and the interpolations worked fine. I then tested to make sure all the points in the first element of recalc had an elevation: they did. I repeated this test with pts, and it returned false.

This implementation was successful. Between the two new developments (only connections needed, every other points), the number of queries for my test route was reduced from 17 to 5, which translates to a huge time advantage.