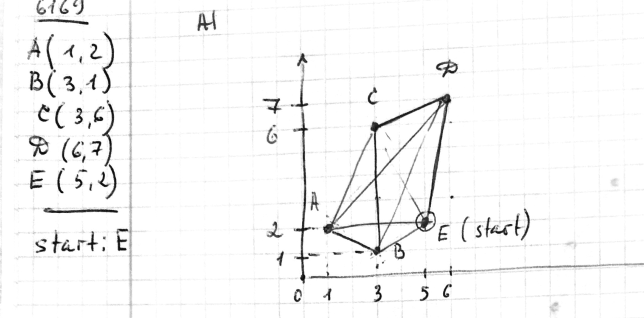
“The Floyd-Warshall algorithm is an example of dynamic programming. The algorithm compares all possible paths through the graph between each pair of vertices” - https://en.wikipedia.org/wiki/Floyd%E2%80%93Warshall\_algorithm.

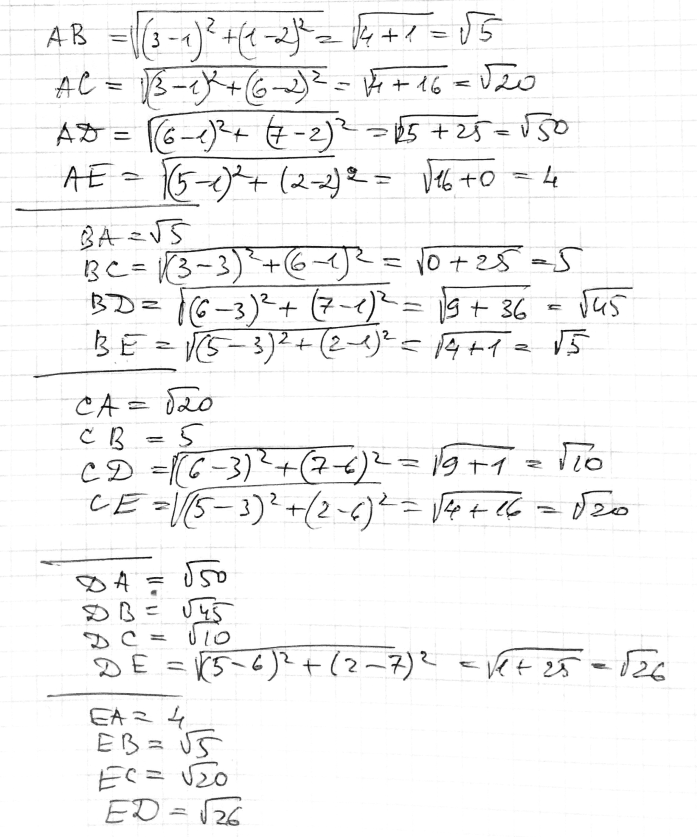
The idea of the algorithm is to find the shortest path between two vertices, for instance 1->2. But we want to ensure that there is no any shorter path from 1 to 2 going through another vertex, for instance, 3. Hence, we check if the path 1->3->2 is shorter than 1->2. Then we repeat our search for all remaining vertices 4 and 5 in our case. I mean we are looking for the shortest path between 1 and 2, trying to reach 2 through vertices 3, 4, 5 accordingly. When we have all distances 1->2, 1->3->2, 1->4->2 and   
1->5->2, we compare and pick the shortest one. The same strategy we use to find the shortest paths between all remaining pairs of vertices - 1->3, 1->4, 1->5 gradually using as mediator remaining vertices. For example for 1->3 it will be 1->2->3, 1->4->3, 1->5->3. Our goal results in the matrix with the shortest distance between all pairs of vertices presented. Next, we eventually can find the whole path from given starting point to the same starting point after going through all vertices by adding the last distance between endpoint and starting point.

Implementation:

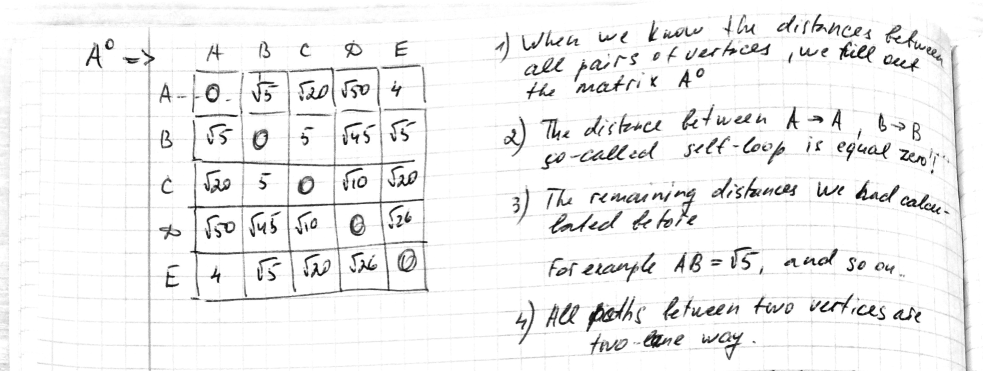
Source: A(1,2) B(3,1) C(3,6) D(6,7) E(5,2). Starting point is E.



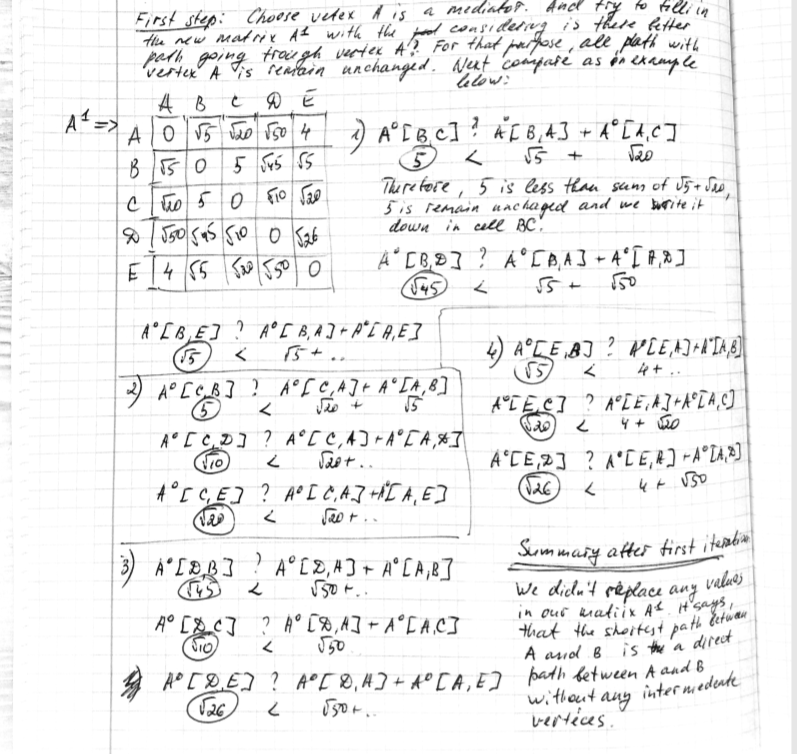
The distances between all pair calculated



Next, we fill in matrix A0 with paths we had got from calculation.



Then, based on values from matrix A0 and using formula A0[i,j] ? A0[I, k] + A0[k,j] we are searching for the shortest path between all pair of vertices BA, BC, BD and BE where the intermediate vertex is A.



If we find that path, for instance, BA+AC is shorter than BC, we will replace the value of BC from the matrix A0 and paste the sum of BA and AC in the cell BC in matrix A1. However, there is no way to meet this condition. Thus, all values from matrix A0 get delivered to matrix A1. To summarize, there is no shorter path between BC, BD, BE, CB, CD, CE, DB,DC,DE, EB,EC,ED than the direct ones.

Next step: we repeat our calculation for the matrix A2 based on values from matrix A1.  
Here is the formula: A1[i,j] ? A1[I, k] + A1[k,j], where k = B;

Then, for matrix A3 : A2[i,j] ? A2[I, k] + A2[k,j], where k = C;

We repeat calculation for remain vertices we have k= D, k = E, building each time a new matrix.

When we go through all pairs of vertices we will get the shortest path between them. In our case, the shortest paths between all pairs of vertices are direct paths between them.

Finally, we calculate the distance from given point to the same point via all vertices on the way. The path will go through the shortest distance between every pairs of vertices.