Algorithm and Languages

I implemented Floyd's Algorithm using 

Why Floyd's?

- I am starting to contract senioritis
- The python Implementation is short & easy

Why Java?

- Familiarity and comfortability
- It is much faster than Python

Java Facts:

- Compiled
- Statically Typed
- Java was originally named Oak

Implementation

```java
// Takes in adjacency Matrix (W)
public static int[][] floyd(int[][] W) {
    // Get the number of vertices in the graph.
    int numVert = W.length;
    // Initialize a 2D array to store the shortest distances.
    int[][] dist = new int[numVert][numVert];
    // Copy the input weights into the distances array.
    for (int i = 0; i < numVert; i++) {
        for (int j = 0; j < numVert; j++) {
            dist[i][j] = W[i][j];
        }
    }
    // Find the shortest paths between all pairs of vertices.
    for (int k = 0; k < numVert; k++) {
        for (int i = 0; i < numVert; i++) {
            for (int j = 0; j < numVert; j++) {
                // If the distance from i to j is greater than the distance from i to k plus
                // the distance from k to j, update the distance from i to j to be the sum of
                // the distances from i to k and from k to j.
                if (dist[i][k] != Integer.MAX_VALUE && dist[k][j] != Integer.MAX_VALUE &&
                        dist[i][j] > dist[i][k] + dist[k][j]) {
                    dist[i][j] = dist[i][k] + dist[k][j];
                }
            }
        }
    }
    // Return the 2D array of shortest distances between all pairs of vertices.
    return dist;
}
```

Results and Conclusion

Java's Runtime ≈ 0.064 sec

Python's Runtime ≈ 7.69 sec

Java's runtime is 120.16 times faster than Python's runtime!

In the future, my preferred language to use would be…

- Java if speed is crucial
- Python for anything else.