```
point[] P, point[] Q, point[] R

Path[P.length][Q.length][R.length]
main{
    first check dupes on list P, Q, R, if so, return false
    testLfeasible(0,0,0)
    return Path[n][n][n]
}
testLfeasible(pi,qi,ri, L)
{
    if(pi == n && qi == n && ri == n)
        return;

    if (max(||P[pi + 1] - Q[qi]||, ||R[ri] - Q[qi]||, ||P[pi + 1] - R[ri]||) =< L)
        path[pi + 1,qi,ri] = true
        testLfeasible(pi + 1,qi,ri,L)
    else
        path[pi + 1,qi,ri] = false

    if (max(||P[pi] - Q[qi + 1]||, ||R[ri] - Q[qi + 1]||, ||P[pi] - R[ri]||) =< L)
        path[pi,qi + 1,ri] = true
        testLfeasible(pi,qi + 1,ri,L)
    else
        path[pi,qi + 1,ri] = false

    if (max(||P[pi] - Q[qi]||, ||R[ri + 1] - Q[qi]||, ||P[pi] - R[ri + 1]||) =< L)
        path[pi,qi,ri + 1] = true
        testLfeasible(pi,qi,ri + 1,L)
    else
        path[pi,qi,ri + 1] = false

    if (max(||P[pi + 1] - Q[qi + 1]||, ||R[ri] - Q[qi + 1]||, ||P[pi + 1] - R[ri]||)
=< L)
        path[pi + 1,qi + 1,ri] = true
        testLfeasible(pi + 1,qi + 1,ri,L)
    else
        path[pi + 1,qi + 1,ri] = false

    if (max(||P[pi] - Q[qi + 1]||, ||R[ri + 1] - Q[qi + 1]||, ||P[pi] - R[ri + 1]||)
=< L)
        path[pi,qi + 1,ri + 1] = true
        testLfeasible(pi,qi + 1,ri + 1,L)
    else
        path[pi,qi + 1,ri + 1] = false

    if (max(||P[pi + 1] - Q[qi]||, ||R[ri + 1] - Q[qi]||, ||P[pi + 1] - R[ri + 1]||)
=< L)
```

```
        path[pi + 1,qi,ri + 1] = true
        testLfeasible(pi + 1,qi,ri + 1,L)
    else
        path[pi + 1,qi,ri + 1] = false

    if (max(||P[pi + 1] - Q[qi + 1]||, ||R[ri + 1] - Q[qi + 1]||, ||P[pi + 1] - R[ri
+ 1]||) =< L)
        path[pi + 1,qi + 1,ri + 1] = true
        testLfeasible(pi + 1,qi + 1,ri + 1, L)
    else
        path[pi + 1,qi + 1,ri + 1] = false
}
```