Exercise 3: Sorting Customer Orders

# Approach & Understanding

In this exercise, I have developed a sorting solution for customer orders based on their **total price**. Sorting helps in quickly identifying high-value orders, which is important for prioritization on an e- commerce platform.

# Code

public class sortdata

{

static class o

{

String iD; String cN; double tP;

o(String iD , String cN , double tP)

{ this.iD= iD ; this.cN = cN ; this.tP= tP ; }

public String toString()

{ return "o [ id = '" + iD + "' , c = '" + cN + "' , tp = " + tP + " ]" ; }

}

public static void b(o[] d)

{

int N = d.length ;

for( int a =0 ; a < N -1 ;a++)

{

for( int b=0 ; b< N - a -1 ;b++)

{

if ( d[b].tP > d[b+1].tP )

{

o tmp = d[b]; d[b] = d[b+1]; d[b+1] = tmp ;

}

}

}

}

public static void quick(o[] x , int L , int H )

{

if ( L < H )

{

int c = cutPoint(x , L , H ); quick(x , L , c -1 );

quick(x , c +1 , H );

}

}

public static int cutPoint(o[] x , int L , int H )

{

double P = x[H].tP ; int U = L - 1 ;

for(int y = L ; y < H ; y++ )

{

if( x[y].tP <= P )

{

U++ ; o t = x[U] ; x[U] = x[y] ; x[y] = t ;

}

}

o t = x[U+1]; x[U+1] = x[H]; x[H] = t ;

return U + 1 ;

}

public static void main(String[] Z)

{

o[] k = {

new o("O001","meena",250.75),

new o("O002","nithya",150.50),

new o("O003","sathya",300.00),

new o("O004","ramu",100.00),

new o("O005","raju",200.25)

};

System.out.println("Base list:");

for (o e : k ) { System.out.println(e); }

o[] B = k.clone(); b(B);

System.out.println("\nBubble sort done :"); for (o e : B ) { System.out.println(e); }

o[] Q = k.clone(); quick(Q , 0 , Q.length -1 );

System.out.println("\nQuick sort done :"); for (o e : Q ) { System.out.println(e); }

}

}

**OUTPUT:**

