Algoritmo

crearCoordenadasRandom

Entrada

Valor N entero

Algoritmo Random

Salida

Secuencia coordenadas de coordenadas

Pseudocodigo

coordenadas<-[n][2]

para i=0 hasta que i<coordenadas.lenght {

coordenadas[i][0]<- Math.abs(random.nextInt()%MAP\_SIZE);

coordenadas[i][1]<- Math.abs(random.nextInt()%MAP\_SIZE);

i<-i++

}

Res<-coordenadas

Algoritmo

costosPorCoordenadas

Entrada

Valor i,j

Salida

Pseudocodigo

Para i=0 hasta que i<coordenadas.lenght{

Para j=0 hasta que j<coordenadas.lenght{

Costo[i][j]<- calcularCostodeViajeentreCiudades(i,j);

j<-j++

}

i<-i++

}

Algoritmo

calcularCostodeViajeentreCiudades

Entrada

Valor i,j

Salida

Valor n de costo de viaje entre ciudades

Pseudocodigo

Int dx <- coordenadas[i][0]-coordenadas[j][0];

Int dy<- coordenadas[i][1]-coordenadas[j][1];

Res<- Math.sqrt(dx\*dx+dy\*dy);

Algoritmo

calcularCosto

Entrada

Secuencia ruta

Algoritmo Random

¿?

Pseudocodigo

Res<- calcularCosto(ruta, false);

Algoritmo

calcularCosto

Entrada

Secuencia ruta

Valor isVerbose boolean

Salida

Valor costodeViaje costo de viaje

Pseudocodigo

costodeViajes<-0

para i=1 hasta que i=>ruta.length {

costodeViaje<-costodeViaje+ costo[ruta[i-1]][ruta[i]];

i++;

Si isVerbose==verdadero{

System.out.println("Costo desde "+ruta[n-1]+" to "+ruta[0]+": "+costo[ruta[n-1]][ruta[0]]);

}

}

costodeViajes <-costodeViaje+ costo[ruta[n-1]][ruta[0]];

si isVerbose ==true{

System.out.println("costo from "+ruta[n-1]+" to "+ruta[0]+": "+costo[ruta[n-1]][ruta[0]]);

}

Res<-costodeViaje

Algoritmo

printRuta

Entrada

Secuencia ruta

Salida

Pseudocodigo

Para i=0 hasta que i=>ruta.lenght {

System.out.print(ruta[i]+" ");

I++

}

Algoritmo

printCosto

Entrada

Salida

Pseudocodigo

System.out.println("matriz de costo para el problema del viajante de comercio:");

Para i=0 hasta que i>=costo.lenght{

Para j=0 hastq que j=>costo[i].lenght{

System.out.print(costo[i][j]+" ");

J++}

I++

System.out.print("\n");