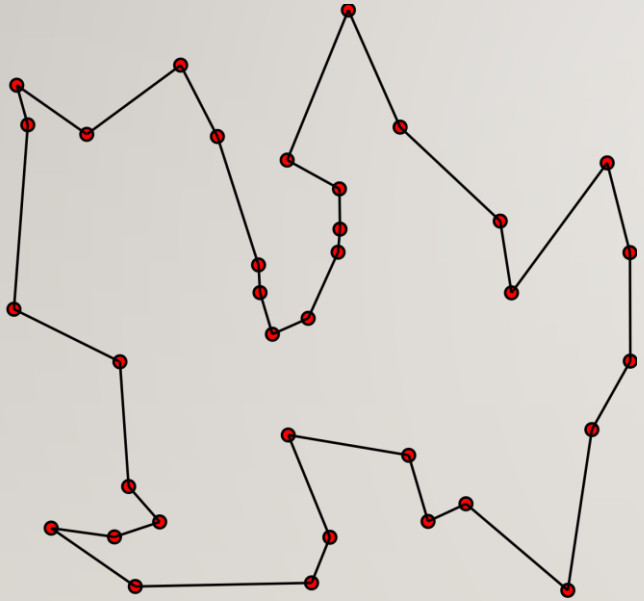
PROGRAMACIÓN DINÁMICA

Javier Ramírez Pulido
Manuel Ángel Rodríguez Segura
Alejandro Ruiz Rodríguez
Ángel Solano Corral

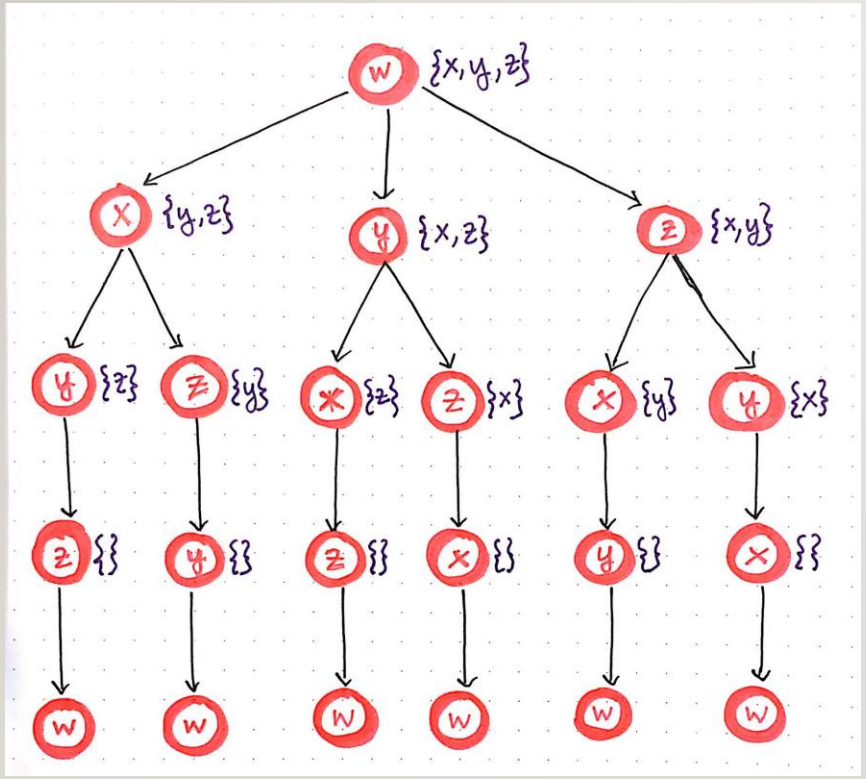
GRUPO 6

DESCRIPCIÓN DEL PROBLEMA



El problema del Viajante de Comercio, tiene como objetivo partir de una ciudad, realizar un recorrido entre diferentes ciudades de forma que no se repita el camino por una de ellas y volver a la de origen con la menor distancia posible

SOLUCIÓN PROPUESTA



Para todos los subproblemas de tamaño n-1 se realizará un desarrollo recursivo que combina todas las soluciones posibles, y de entre ellas, elige la mejor en cada nivel.

PSEUDOCÓDIGO

```
DistanciaYRecorridoMinimo(nodo_inicio, nodos_por_pasar, ciudadesRecogidas, nodo_empece)
Variable global
ENTERO MAX = 2147483647;            //mayor valor que puede tomar un entero
Variables locales
ENTERO másCorto = MAX;
ENTERO distancia, elementoBorrado, indiceMasCorto;
VECTOR<CIUDAD> comprobaciones;
PAIR <INT, VECTOR<INT>> solución, aux;

principio
    si nodos_por_pasar = ∅ entonces
        solución.first <- calcularDistanciaCiudades(1, nodo_inicio);
        solución.second <- insertar(nodo_inicio);
        devuelve solución;
    sino
    para todo j en nodos_por_pasar hacer

        comprobaciones <- insertar(nodo_incio, nodos_por_pasar(j));
        elementoBorrado <- nodos_por_pasar(j);
        nodos_por_pasar <- borrar(j);

        aux <- DistanciaYRecorridoMinimo(elementoBorrado, nodos_por_pasar, ciudadesRecogidas, nodo_empece) ;

        distancia:= calcularDistanciaCiudades(comprobaciones) + aux.first;
        nodos_por_pasar <- insertar(elementoBorrado);
        comprobaciones.vaciar;

        si distancia<masCorto entonces

            solución <- aux;
            solución.first <- distancia;
            masCorto <- distancia ;
            indiceMasCorto <- nodo_inicio

        fsi
    fpara;
    solución.first <- masCorto;
    solución.second <- indiceMasCorto;
    fsi
fsi
devuelve solución;
fin
```

Función recursiva que devuelve la solución al problema, es decir, el recorrido óptimo de ciudades que se proporcionan mediante un fichero de .tsp así como la distancia total del mismo

CASOS DE EJECUCIÓN (PARA 4 CIUDADES)

1

```
**********************************************************************************************************
                                        DISTANCIAS DIRECTAS
**********************************************************************************************************

             -------------------------------
             1       2       3       4
             -------------------------------
   Ciudad 1  |0      6       5       4
   Ciudad 2  |6      0       1       4
   Ciudad 3  |5      1       0       4
   Ciudad 4  |4      4       4       0

**********************************************************************************************************

**********************************************************************************************************

**********************************************************************************************************

                                    TODOS LOS CAMINOS POSIBLES

                                              1
                                            / | \
                                          /   |   \
                                        /     |     \
                                      /       |       \
                                    2         3         4
                                  / \       / \       / \
                                 3   4     2   4     2   3
                                 |   |     |   |     |   |
                                 4   3     4   2     3   2
                                 |     \   \   /   /     |
                                 \___    \   \ /   /    ___/
                                     \\_  |  _//
                                        \ | /
                                          1
```

2

```
**********************************************************************************************************
                                              *1*
                                            / | \
                                          /   |   \
                                        /     |     \
                                      /       |       \
                                    *2*       3         4
                                  / \       / \       / \
                                *3*  4     2   4     2   3
                                 |   |     |   |     |   |
                                *4*  3     4   2     3   2
                                 |     \   \   /   /     |
                                 \___    \   \ /   /    ___/
                                     \\_  |  _//
                                        \ | /
                                          *1*
Cogiendo el camino marcado con '*' la longitud es: 15
**********************************************************************************************************
                                              *1*
                                            / | \
                                          /   |   \
                                        /     |     \
                                      /       |       \
                                    *2*       3         4
                                  / \       / \       / \
                                3   *4*2    4   2     3   \
                                |   |     |   |     |   |
                                4   *3*4    2   3     2   |
                                 |     \   \   /   /     |
                                 \___    \   \ /   /    ___/
                                     \\_  |  _//
                                        \ | /
                                          *1*
Cogiendo el camino marcado con '*' la longitud es: 19
```

3

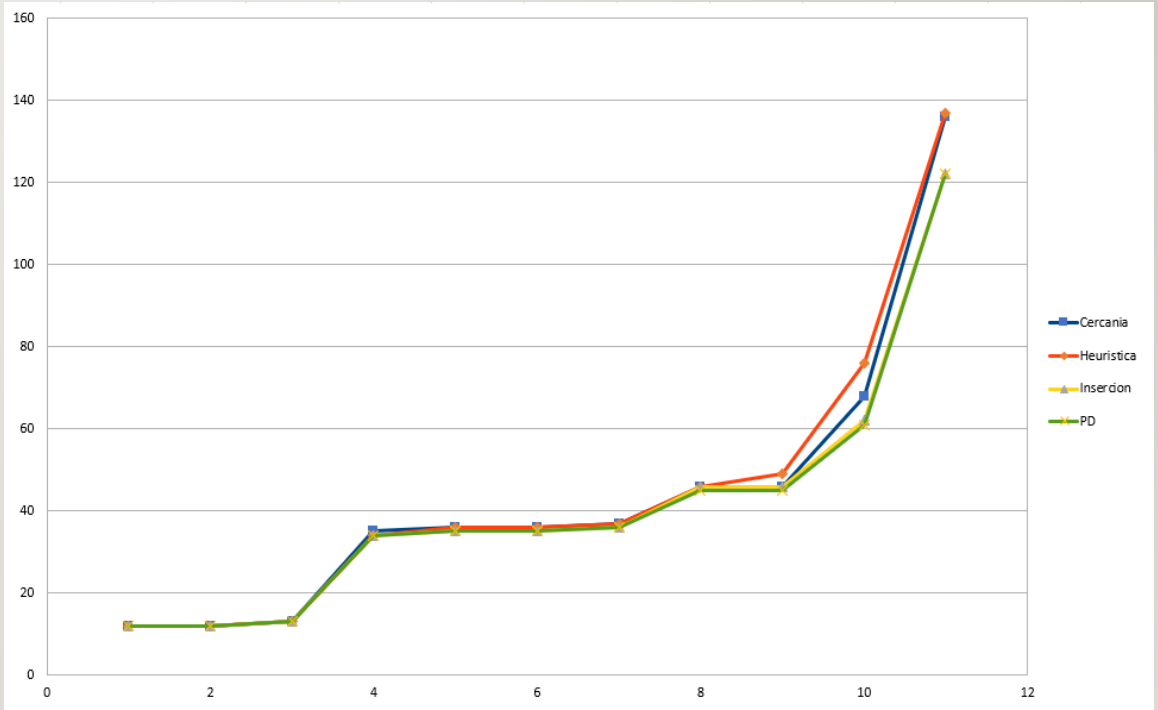

1
2 *3* 4
3 4*2* 4 2 3
4 3*4* 2 3 2
1
ogiendo el camino marcado con '*' la longitud es: 14
1
2 *3* 4
3 4 2 *4*2 3
4 3 4 *2*3 2
1
ogiendo el camino marcado con '*' la longitud es: 19

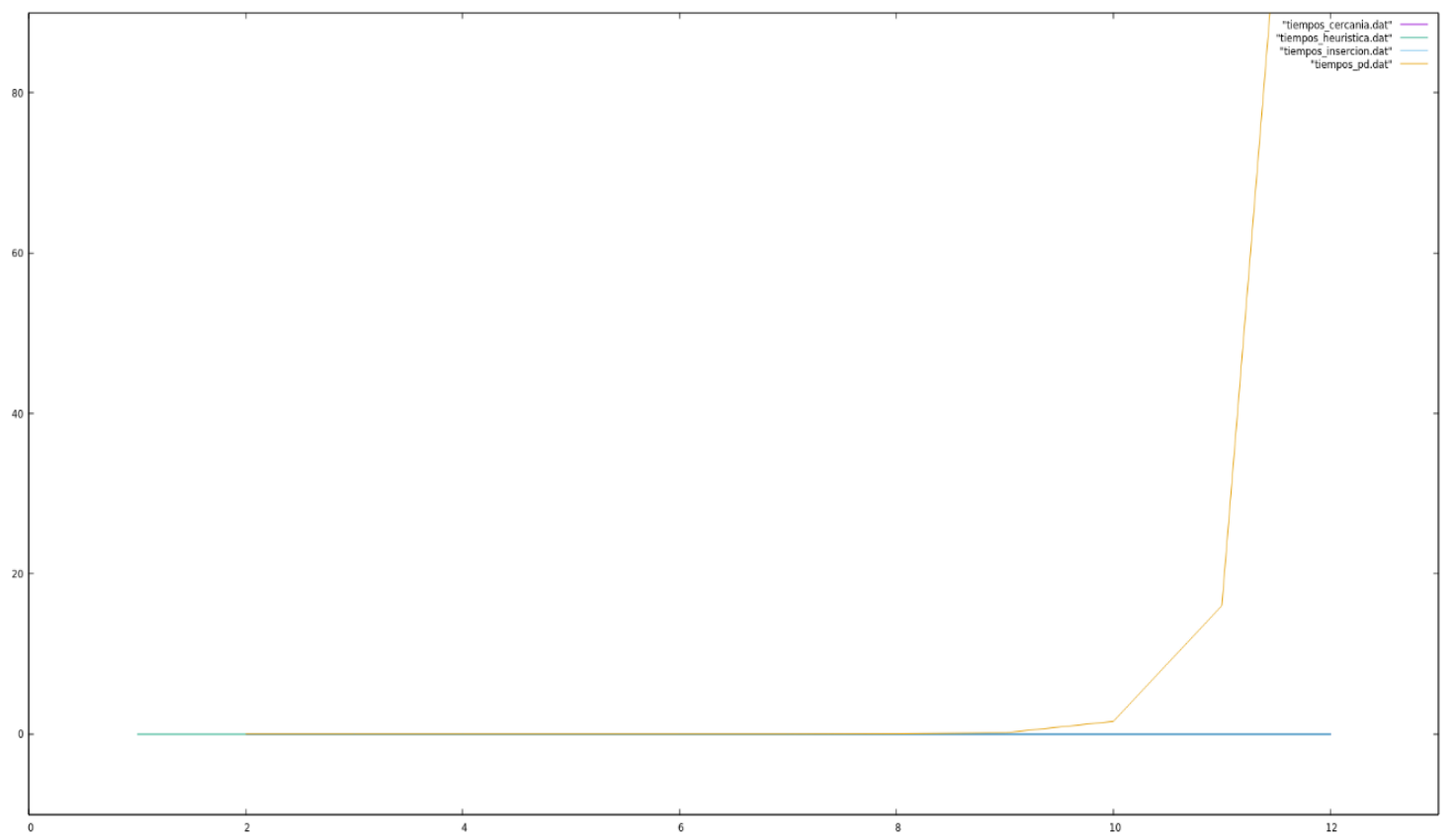
4


1
2 3 *4*
3 4 2 4*2* 3
4 3 4 2*3* 2
1
ogiendo el camino marcado con '*' la longitud es: 14
1
2 3 *4*
3 4 2 4 2 *3*
4 3 4 2 3 *2*
1
ogiendo el camino marcado con '*' la longitud es: 15
A DISTANCIA MAS CORTA MIDE: 14
L RECORRIDO FINAL ES : 1 -> 3 -> 2 -> 4 -> 1

COMPARACIÓN DE SOLUCIONES Y EFICIENCIAS CON ALGORITMOS VORACES (P3)

	TSP CERCANIA	TSP HEURISTICA	TSP INSERCION	TSP PD
a2.tsp	12	12	12	12
a3.tsp	12	12	12	12
a4.tsp	13	13	13	13
a5.tsp	35	34	34	34
a6.tsp	36	36	35	35
a7.tsp	36	36	35	35
a8.tsp	37	37	36	36
a9.tsp	46	46	46	45
a10tsp	46	49	46	45
a11.tsp	68	76	62	61
a12.tsp	136	137	122	122

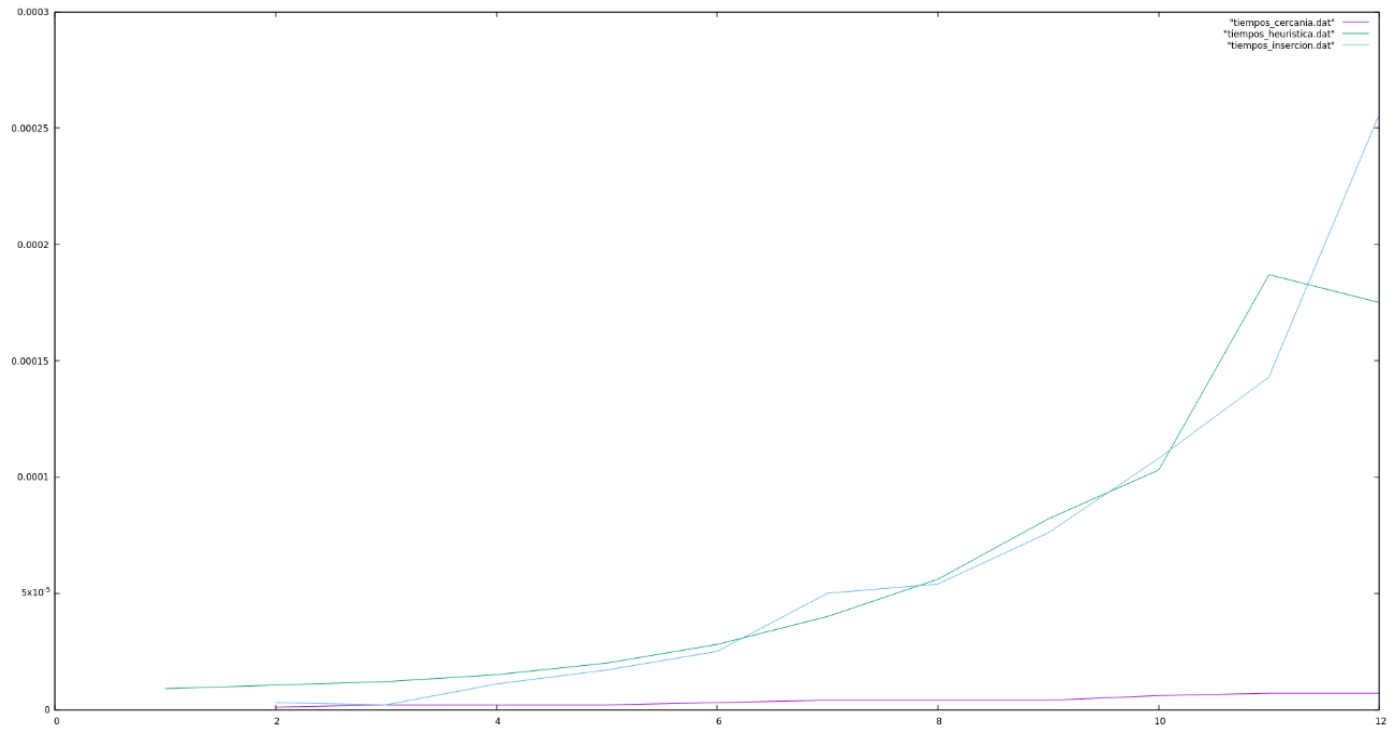




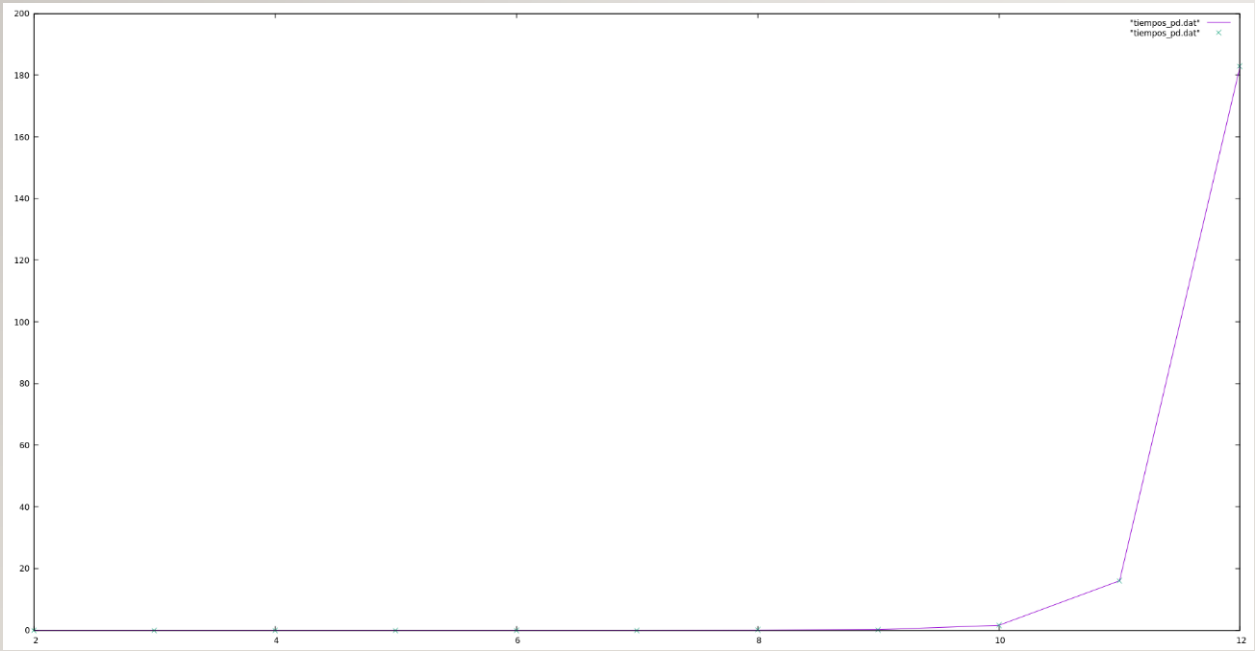
Tiempos entre los diferentes algoritmos (voraces de la P3) y el algoritmo calculado con programación dinámica de esta práctica

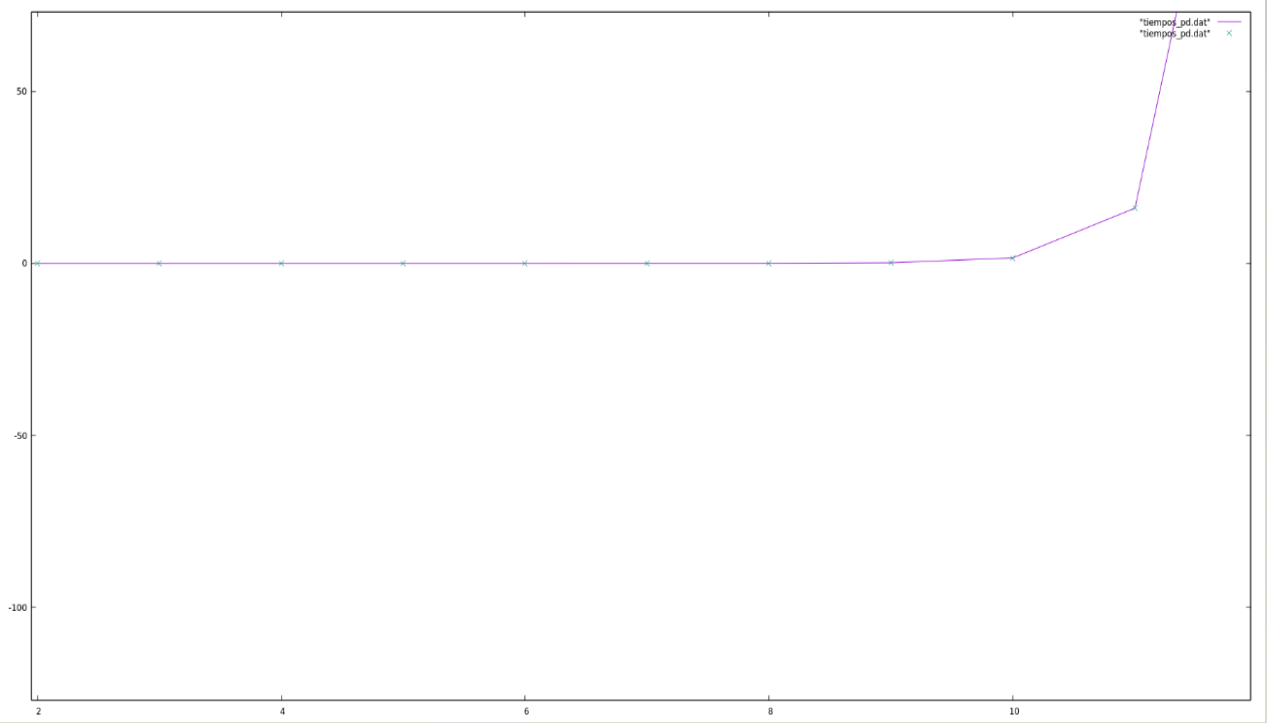
Vamos a verlas más de cerca….

Diferencia de tiempos entre algoritmos voraces (P3)



Tiempos obtenidos en programación dinámica (más detalle)

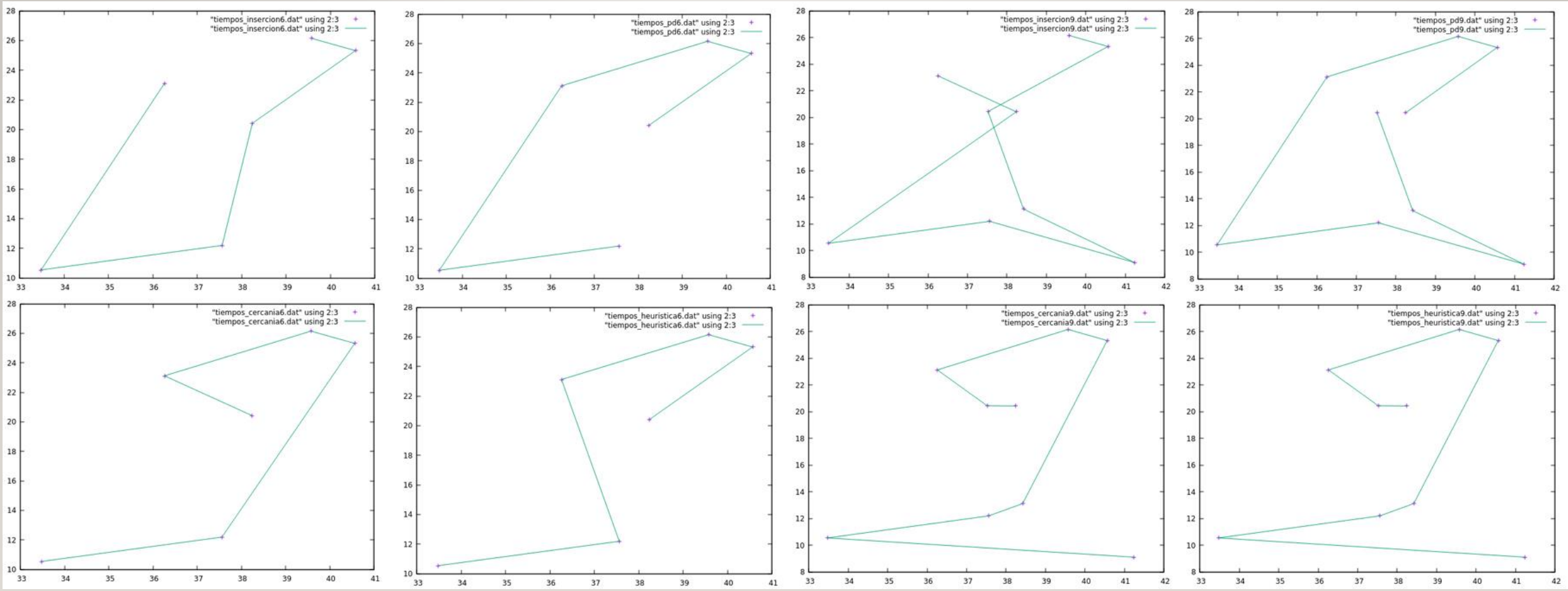




DIFERENCIA DE RECORRIDOS ENTRE ALGORITMOS (FORMA GRÁFICA)

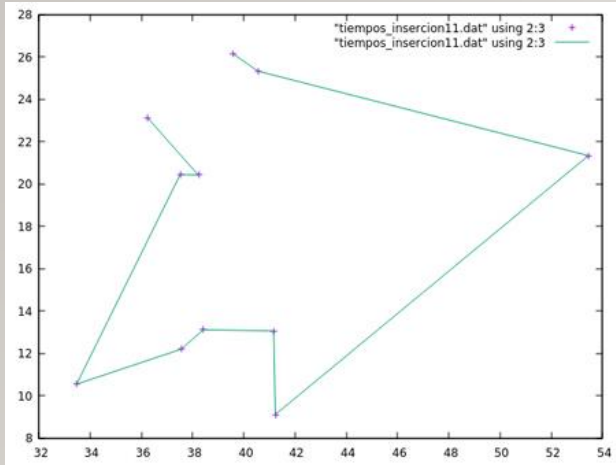
6 ciudades

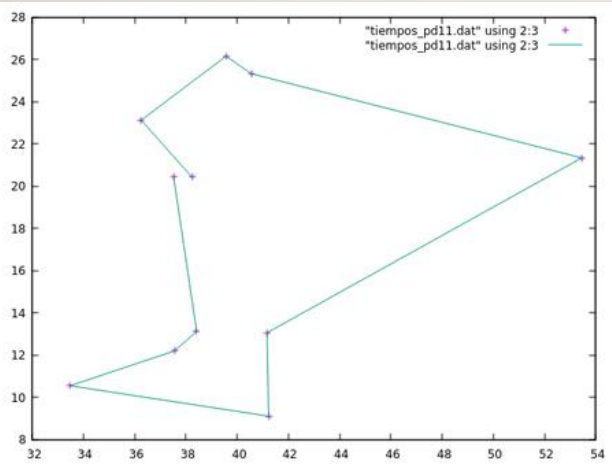
9 ciudades

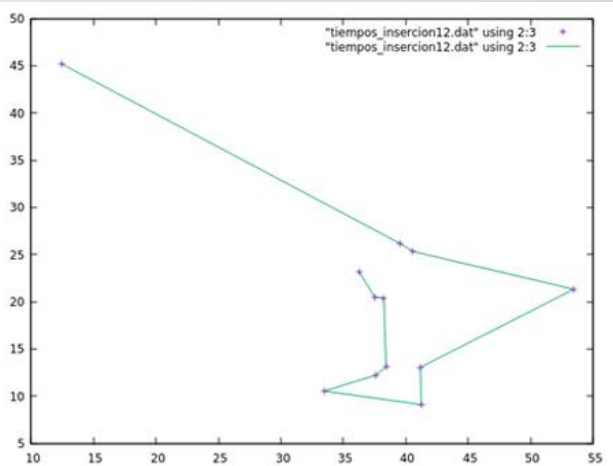


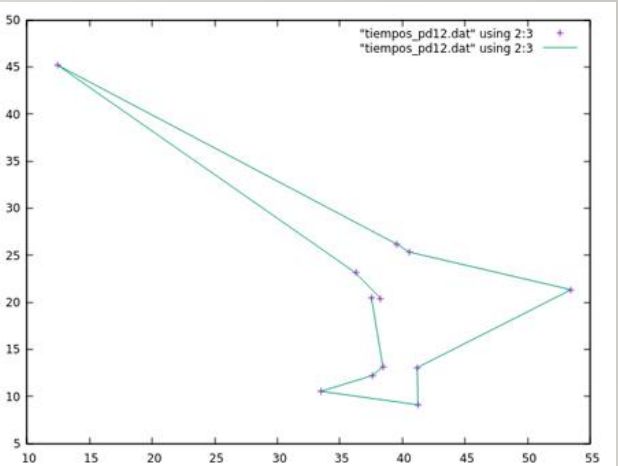
11 ciudades

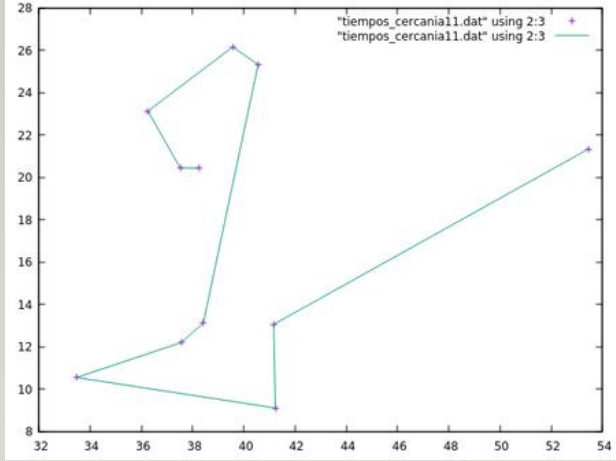
12 ciudades

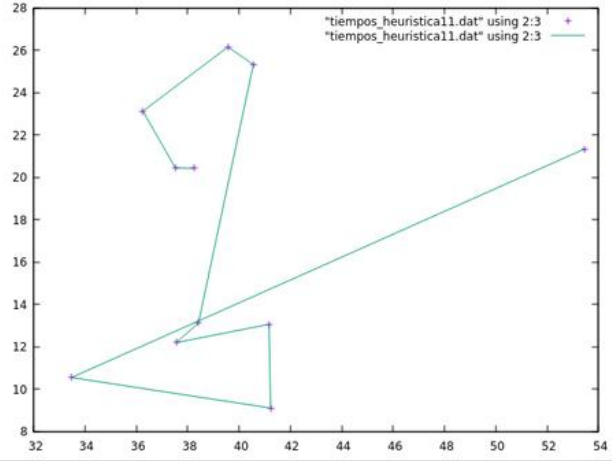


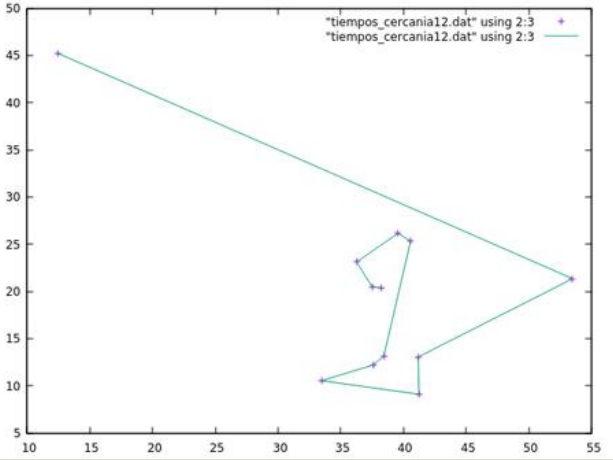


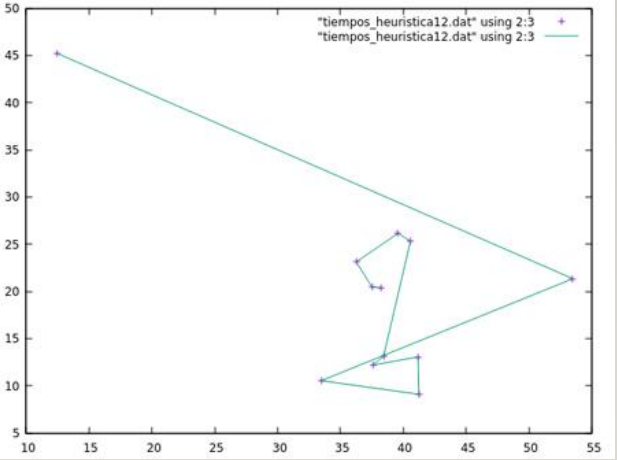












PROGRAMACIÓN DINÁMICA

Javier Ramírez Pulido
Manuel Ángel Rodríguez Segura
Alejandro Ruiz Rodríguez
Ángel Solano Corral

GRUPO 6