Práctica 5: Algoritmos de Vuelta Atrás(Backtracking) y de Ramificación y Poda(Branch and Bound)

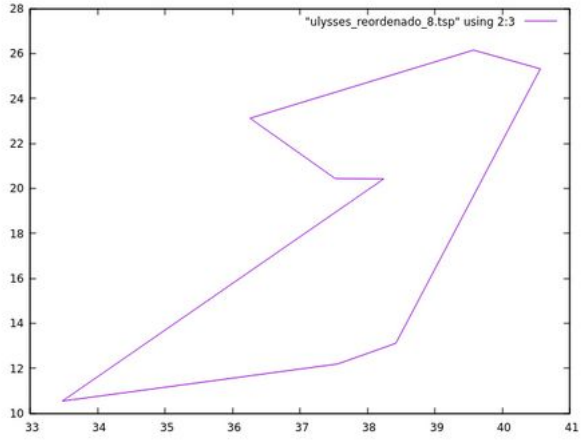
Grupo 4
Integrantes: Raúl Rodríguez Pérez, Francisco Javier Gallardo Molina, Inés Nieto Sánchez, Antonio Lorenzo Gavilán Chacón

Pseudocódigo TSP: Backtracking y Branch and Bound

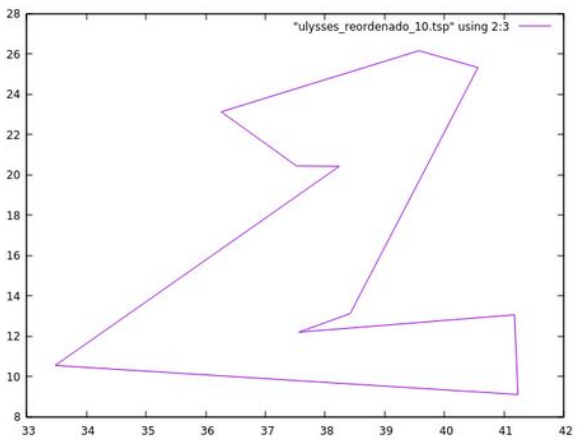
```
void backtracking(ciudad_actual, visitadas, matriz_distancias, cota_real, cota_opt, min_cota, sol_final)
{

        visitadas.add(ciudad_actual)   //añadimos la primera ciudad

        if(visitadas == numero_ciudades){   //caso base
                for(...)
                  distancia += matriz_distancia[][];  //calculamos la distancia del recorrido de ciudades
                                                         visitadas
                if(cota_global > distancia){
                        //actualizamos la distancia global con el valor de distancia
                        //actualizamos sol_final con el recorrido de ciudades de visitadas
                }
        }
        else{
                cota_local = cota_real + cota_opt      //calculamos la cota local
                if(cota_local < cota_global){
                        for(...; i < numero_ciudades; ...){
                                if('i' no se encuentra en 'visitadas'){
                                        //calculamos la cota real
                                        //calculamos la cota optimista
                                        backtracking(i, ..., ..., ...) //volvemos a llamar recursivamente
                                                                          a la funcion
                                        visitadas.pop_back()
                                }
                        }
                }
        }
}
```

```
Algoritmo B&B (matriz, arista_menor, tama) {
    priority_queue<Solucion> Q;
    Solucion n_e, mejor_solucion;
    mejor_solucion.Greedy(matriz);
    CG = mejor_solucion.getDistancia();
    Q.push(n_e);
        while (!Q.empty && Q.top().cota_local() < CG) {
            n_e = Q.top();
            aux = n_e.resto_ciudades();
            for (int i = 0; i < aux.size(); i++) {
                n_e.añadir_ciudad (aux[i], matriz);
                n_e.quitarCiudadRestante (aux[i]);
                if (n_e.Es_Solucion()) {
                    distancia_actual = n_e.Evalua();
                    if (distancia_actual < CG) {
                        CG = distancia_actual;
                        mejor_solucion = n_e;
                    }
                else {
                    if (n_e.Cota_Local < CG)
                        Q.push(n_e);
                }
                n_e.quitarCiudad (matriz);
                n_e.añadirCiudadRestante (aux[i]);
            }
        }
    return mejor_solucion;
}
```

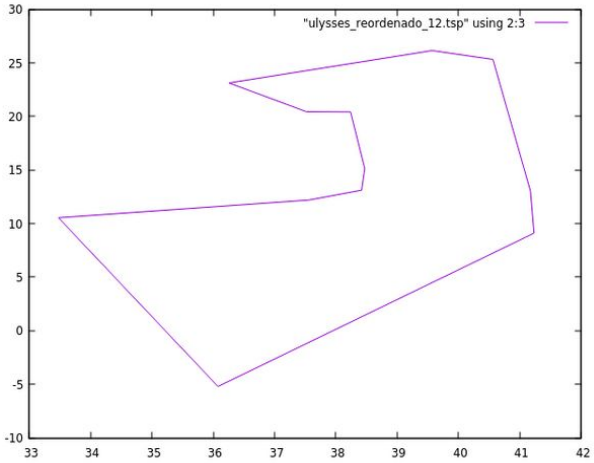
TSP: Recorridos Ulysses 8, 10 y 12



Ulysses 8



Ulysses 10



Ulysses 12

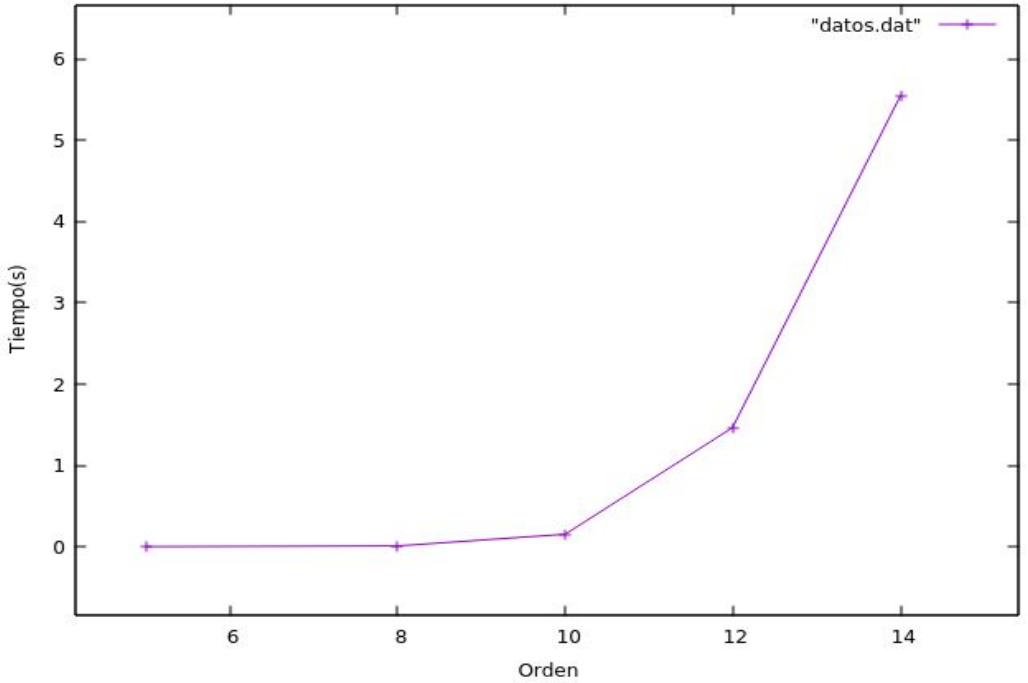
TSP: Tabla comparativa B&B y Backtracking

	Backtracking	Branch and Bound
Ulysses6	199	99
Ulysses7	778	331
Ulysses8	3395	1869
Ulysses9	16640	9219
Ulysses10	57217	33426
Ulysses11	282268	194659
Ulysses12	1310853	893537
Ulysses13	8068748	5391542
Ulysses14	56058989	38452570

Transporte de mercancías

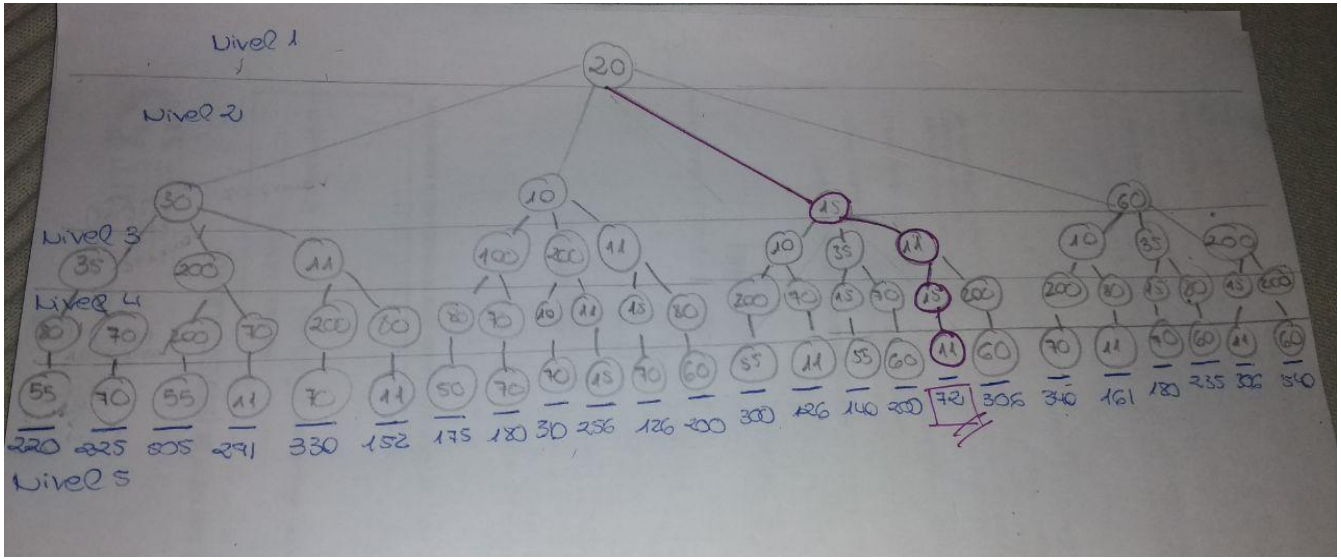
Esquema Recursivo

```
void back_recursivo(Solucion & Sol, int k)
{
if ( k == Sol.size())
   Sol.ProcesaSolucion();
else {
   Sol.IniciaComp(k);
   Sol.SigValComp(k);
   while (!Sol.TodosGenerados(k) {
     if (Sol.Factible(k))
         back_recursivo(Sol, k+1);
     Sol.SigValComp(k);
    }
  }
}
```



Caso de ejecución

20	500	100	40	50
500	30	10	15	60
100	10	35	200	11
40	15	200	80	70
50	60	11	70	55




```
Punto de distribución: 0 ---> Punto de venta: 0
Punto de distribución: 1 ---> Punto de venta: 3
Punto de distribución: 2 ---> Punto de venta: 4
Punto de distribución: 3 ---> Punto de venta: 1
Punto de distribución: 4 ---> Punto de venta: 2
```

```
Punto de distribución inicial 0 distancia minima: 72
Punto de distribución inicial 1 distancia minima: 577
Punto de distribución inicial 2 distancia minima: 191
Punto de distribución inicial 3 distancia minima: 126
Punto de distribución inicial 4 distancia minima: 126
```