Práctica 4

PROGRAMACIÓN DINÁMICA

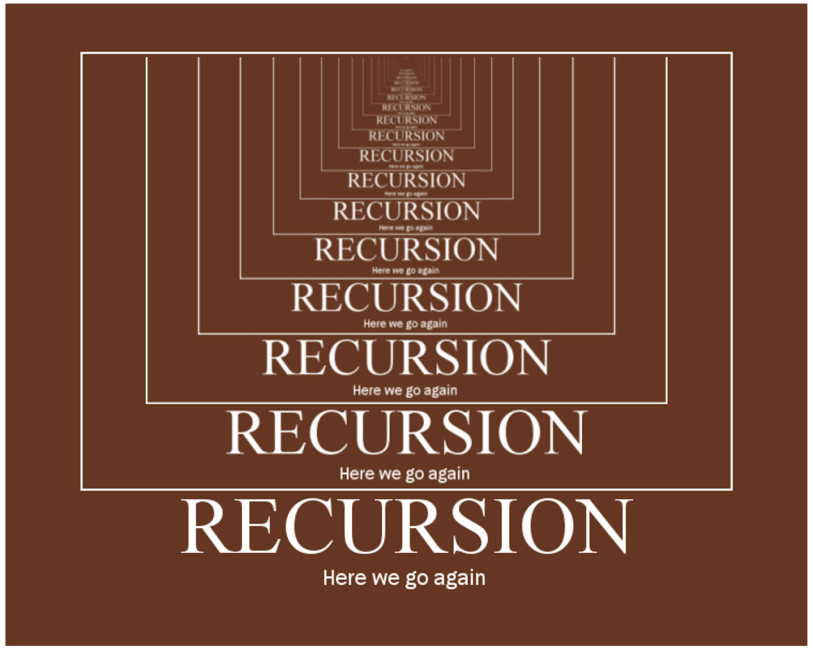
Realizado por Victor Rubia López

CONTENIDO

PROGRAMACIÓN DINÁMICA

DESCRIPCIÓN GENERAL

1. **Resolver problemas recursivamente**
2. **Los problemas los dividimos en subproblemas**
3. **Memoization**

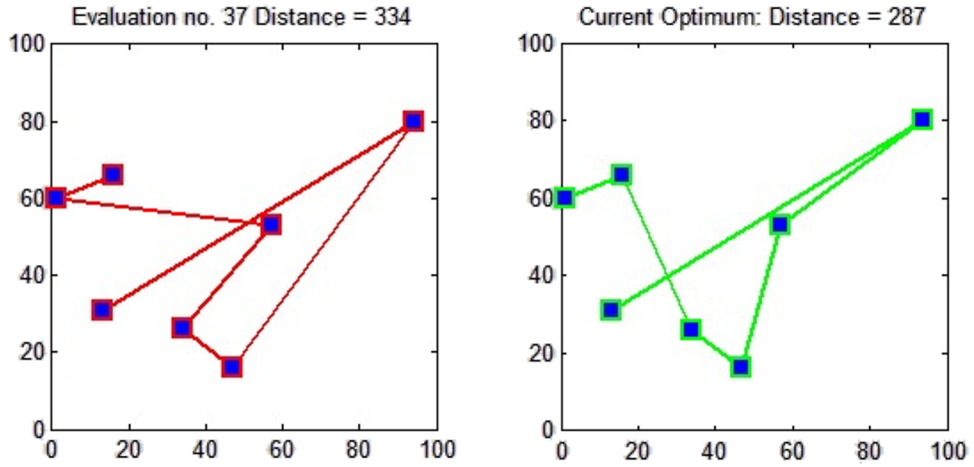


PROBLEMA DEL VIAJANTE DE COMERCIO

ENFOQUE BASADO EN PROGRAMACIÓN DINÁMICA

1. **Recursivamente ciudad con menor coste**

2. **Solución es el camino con menor coste**



PSEUDOCÓDIGO

PROBLEMA DEL VIAJANTE DE COMERCIO BASADO EN PROGRAMACIÓN DINÁMICA

BITMASK
6 CIUDADES

1	0	0	0	0	0

BITMASK
ESTADO INICIAL

```
función algoritmo_pd(bitmask, actual, subproblema, distancias) devuelve sol
variables camino, resultado,i:sol
principio
      si bitmask == (1 << distancias.size-1 entonces devuelve par(camino,
distancias[actual][0])

      si subproblema[bitmask-1][actual].second != -1
                  entonces devuelve subproblema[bitmask-1, actual]

      variable min:= INT_MAX

      para todo j en distancias.size hacer

            variable res:=bitmask & (1 << i)
            si res == 0
                  entonces
                        variable nuevoBitmask:= bitmask | (1 << i)
                        variable nuevoPar:= algoritmo_pd(nuevoBitmask, i, subproblema,
distancias)
                        variable nuevoMin:= distancias[actual][i] + nuevoPar.second
                        variable nuevoPath(nuevoPar.first)
                        nuevoPath.push_back(actual)

                        si nuevoMin < min
                              entonces
                                    min:= nuevoMin
                                    camino:= nuevoPath
                        fsi
            fsi
      fpara

      subproblema[bitmask-1][actual].second:= min
      subproblema[bitmask-1][actual].first:= camino

      devuelve sol:=make_pair(camino, min);
fin
```

BITMASK
6 CIUDADES

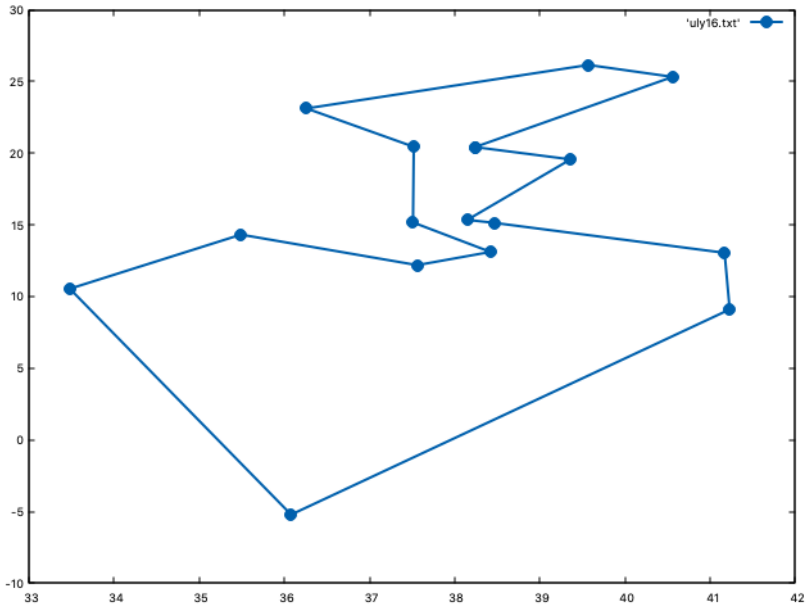
1	1	1	1	1	1

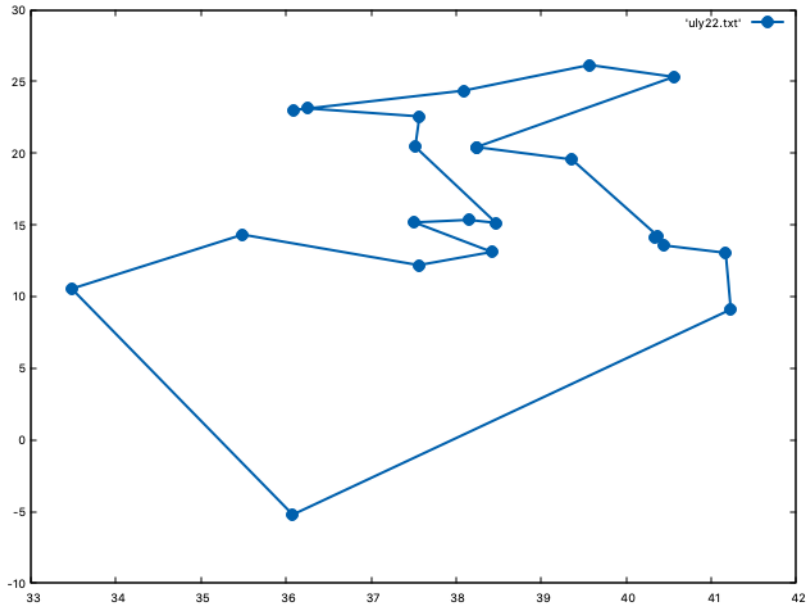
BITMASK
ESTADO FINAL

ALGUNOS ESCENARIOS DE EJECUCIÓN

Ulysses16

Ulysses22





ULYSSES16:

0->2->1->3->7->13->6->5->14->4->10->8->9->11->12->15->0

El coste del camino es: 71

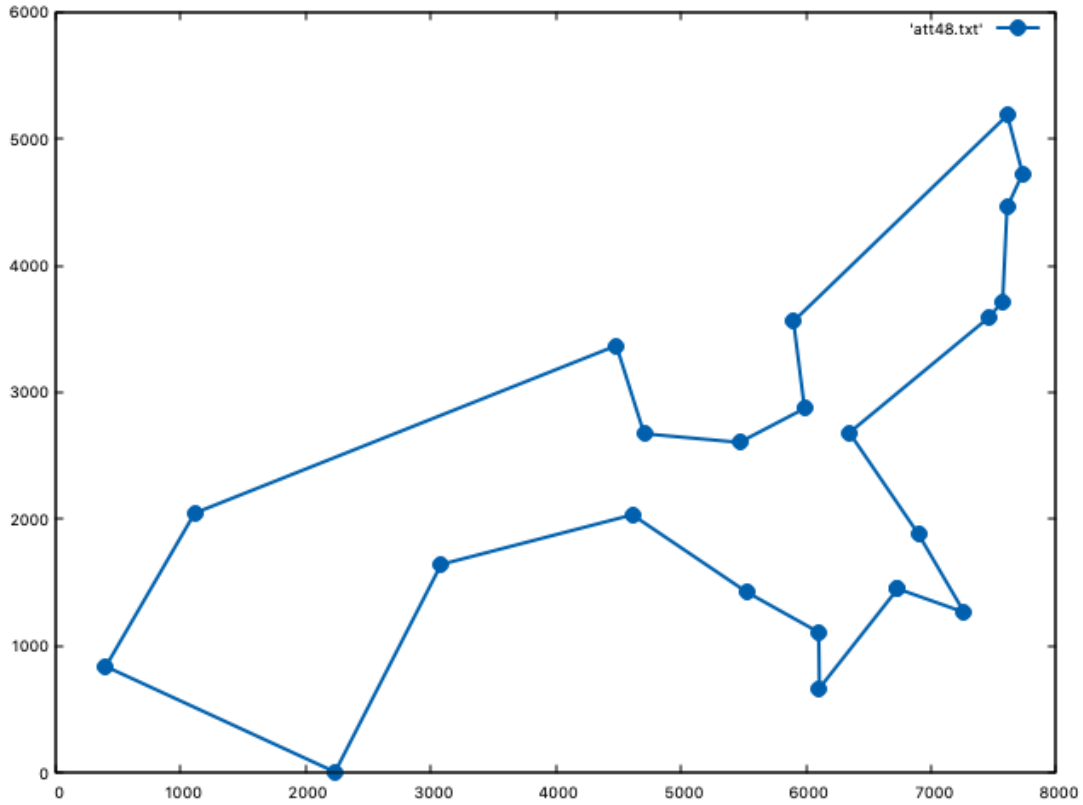
ULYSSES22:

1 -> 8 -> 16 -> 22 -> 4 -> 17 -> 2 -> 3 -> 18 -> 12 -> 14 -> 13 -> 7 -> 6 -> 15 -> 5 -> 9 -> 10 -> 19 -> 20 -> 11 -> 21 -> 1

El coste del camino es: 98

ALGUNOS ESCENARIOS DE EJECUCIÓN

att48



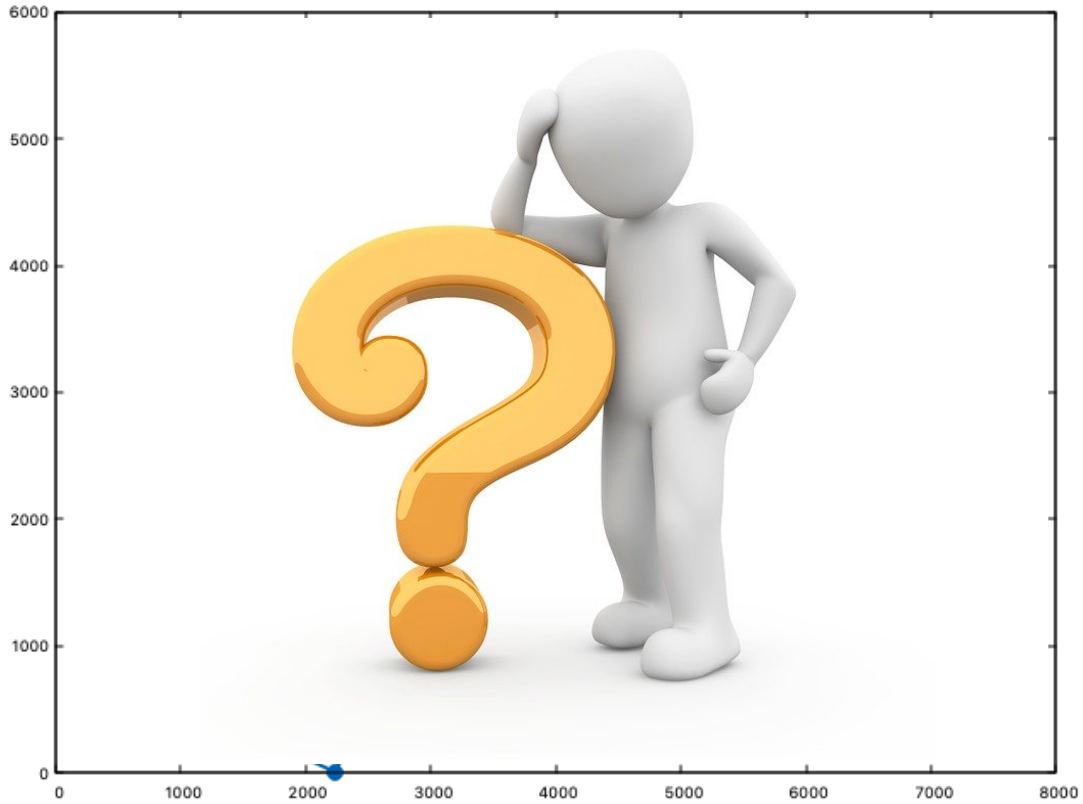
att48 hasta el nodo 22:

0->7->8->14->17->6->5->18->16->19->11->10->12->20->9->3->1->4->13->2->21->15->0

El coste del camino es: 24154

ALGUNOS ESCENARIOS DE EJECUCIÓN

a280



a280:

No es posible calcularlo

VIAJANTE DE COMERCIO. PD.

COMPARACIÓN TIEMPOS OBTENIDOS

	TIEMPO				
	CERCANÍA	INSERCIÓN	INTERCAMBIO	PROGRAMACIÓN DINÁMICA*	PORCENTAJE DE TIEMPOS
Ulysses16	0,00005	0,000056	0,00005	0,892883	178576600%
Ulysses22	0,000014	0,000033	0,00015	120,433	86023571428,6%
att48*	0,000027	0,000081	0,000963	128,052	47426666666,7%
a280	0,000449	0,001413	0,224771		
*= hasta 22 nodos					

*Resultados obtenidos en computador con i7-8700B

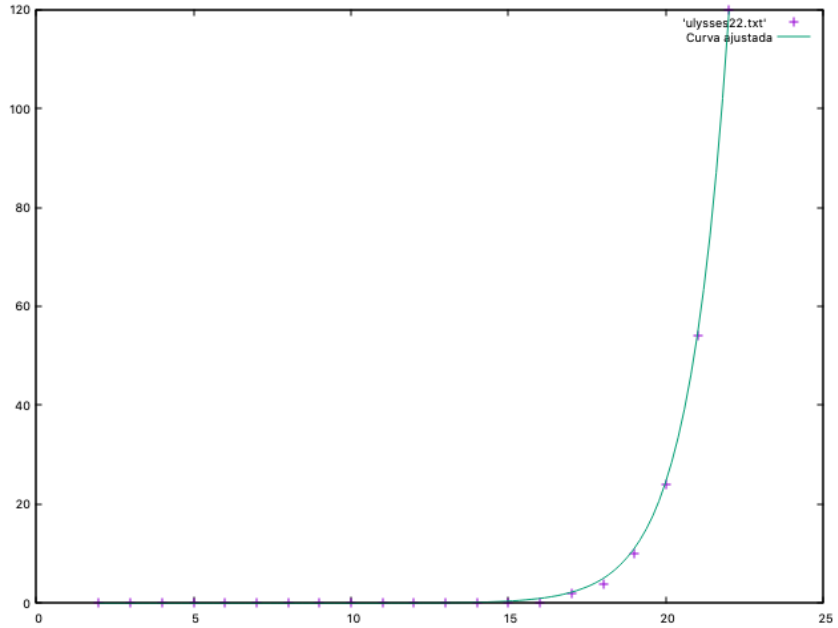
VIAJANTE DE COMERCIO. PD.

TIEMPOS OBTENIDOS. FUNCIÓN DE EFICIENCIA.

ulysses22

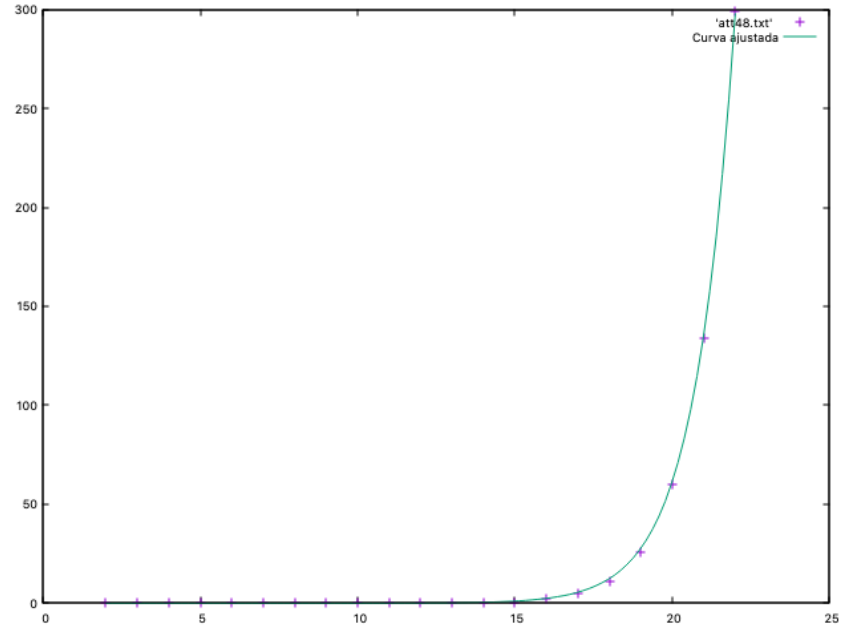
$O(n^2 2^n)$

att48



$$f(x) = 0{,}000102726 \cdot x^2 \cdot 0{,}000573007 \cdot 2^x$$

```
Asymptotic Standard Error
=========================
+/- 1.608e+07    (1.565e+13%)
+/- 8.97e+07     (1.565e+13%)
```



$$f(x) = 0{,}000228646 \cdot x^2 \cdot 0{,}000641378 \cdot 2^x$$

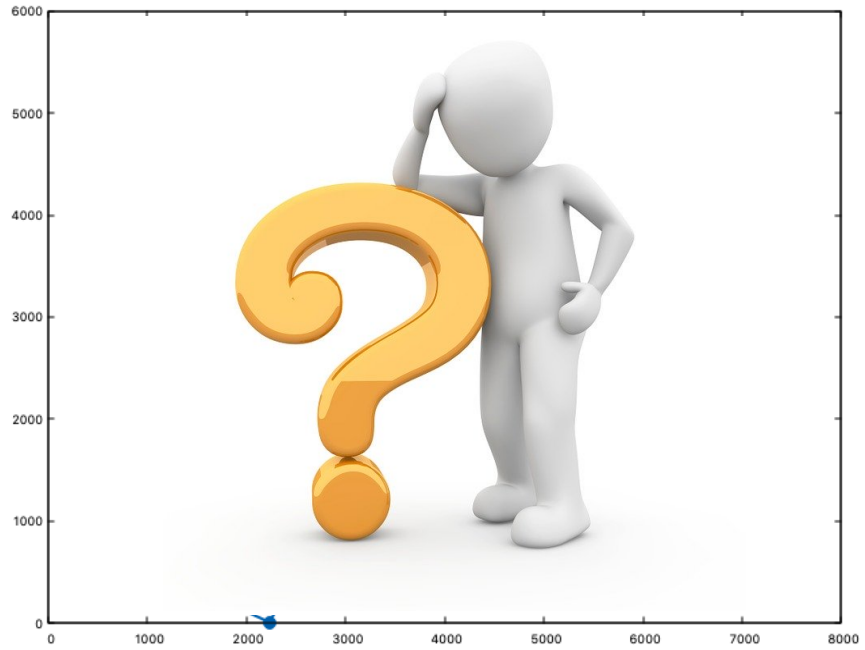
```
Asymptotic Standard Error
=========================
+/- 7.539e+06    (3.297e+12%)
+/- 2.115e+07    (3.297e+12%)
```

*Resultados obtenidos en computador con i7-8700B

VIAJANTE DE COMERCIO. PD.

TIEMPOS OBTENIDOS. FUNCIÓN DE EFICIENCIA.

a280



$f(x) = ?$

Basándonos en ***att48***

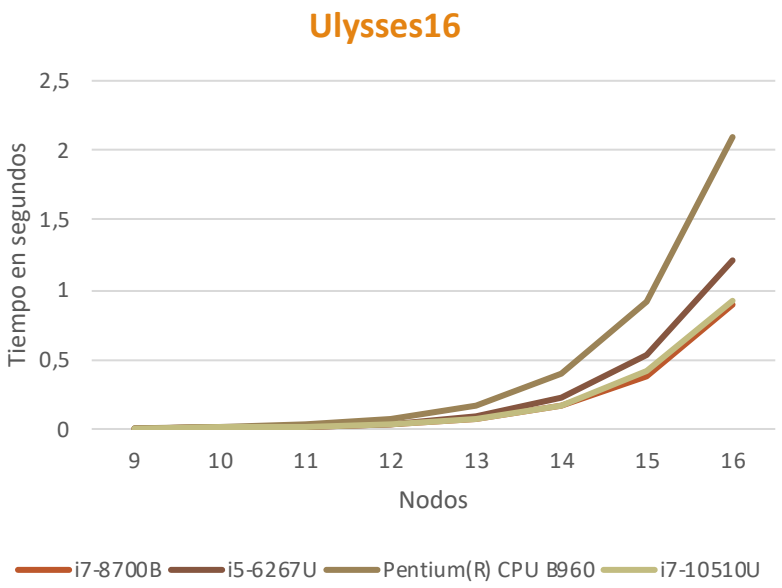
$$f(280) = 0{,}000228646 \cdot 280^2 \cdot 0{,}000641378 \cdot 2^{280}$$

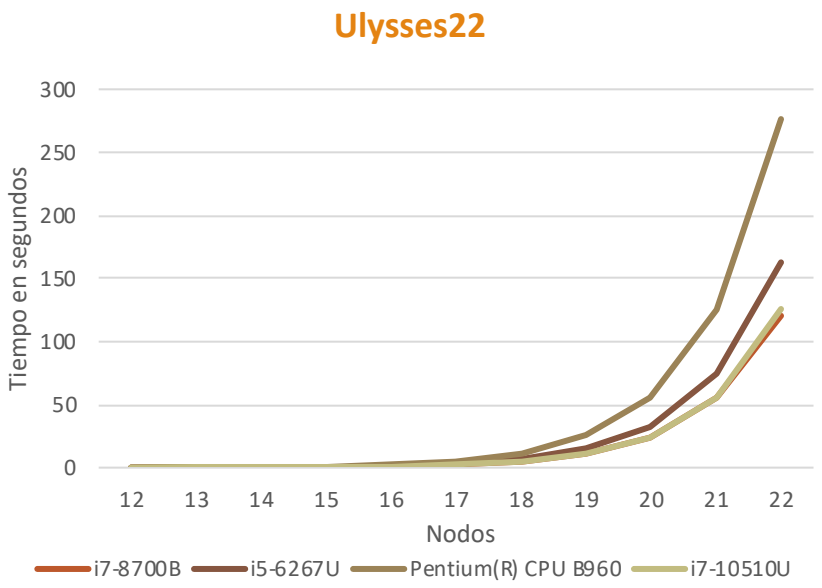
$$4612288267355320 0 \text{ siglos}$$
$$\text{en computarlo}^*$$

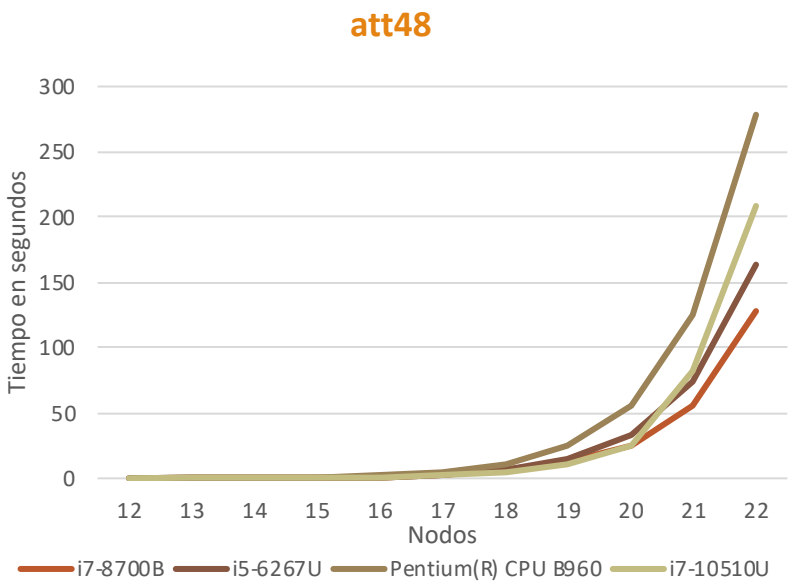
*Resultados obtenidos en computador con i7-8700B

VIAJANTE DE COMERCIO. PD.

TIEMPOS OBTENIDOS. COMPARATIVA EN DISTINTOS COMPUTADORES







VIAJANTE DE COMERCIO. PD.

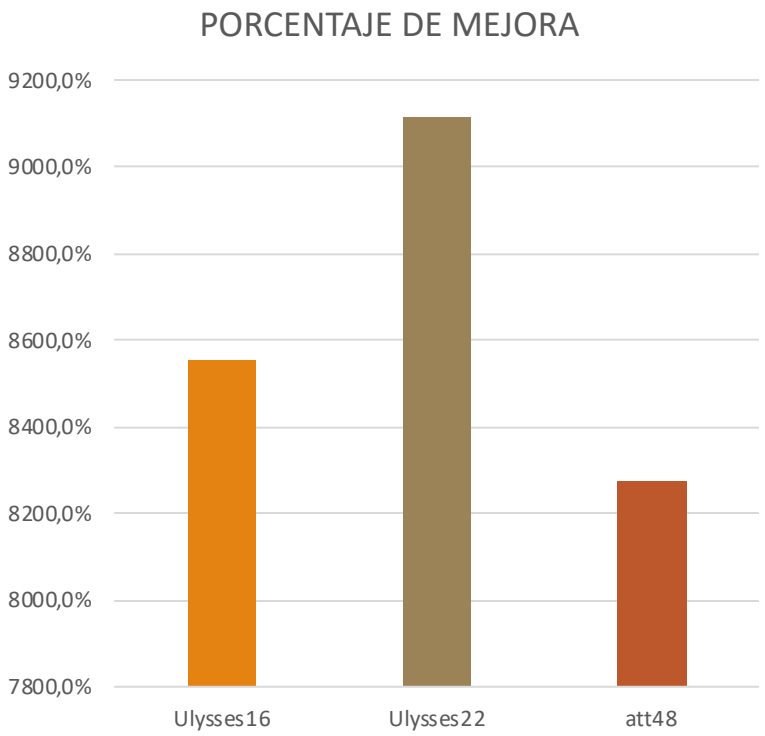
COMPARACIÓN COSTES OBTENIDOS

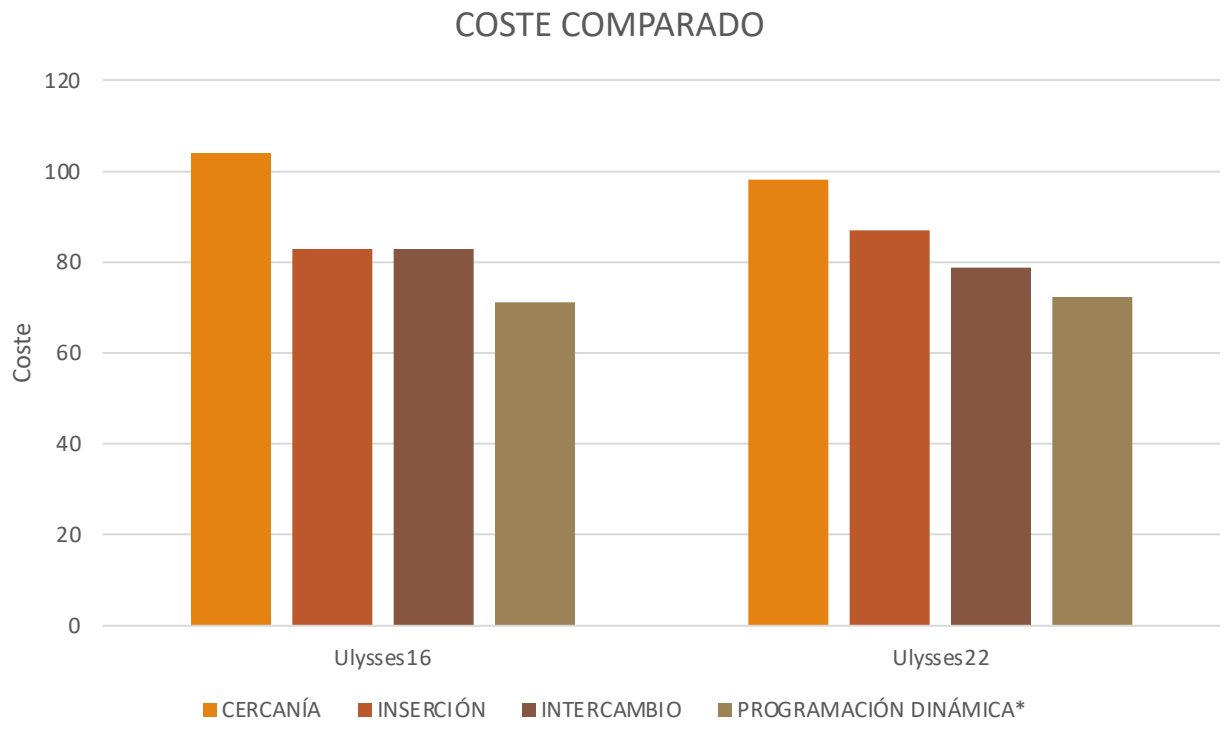
	COSTE				
	CERCANÍA	INSERCIÓN	INTERCAMBIO	PROGRAMACIÓN DINÁMICA*	PORCENTAJE DE MEJORA
Ulysses16	104	83	83	71	8554,2%
Ulysses22	98	87	79	72	9113,9%
att48*	29346	30678	29198	24154	8272,5%
a280					
*= hasta 22 nodos					

*Resultados obtenidos en computador con i7-8700B

VIAJANTE DE COMERCIO. PD.

COMPARACIÓN COSTES OBTENIDOS. GRÁFICAS

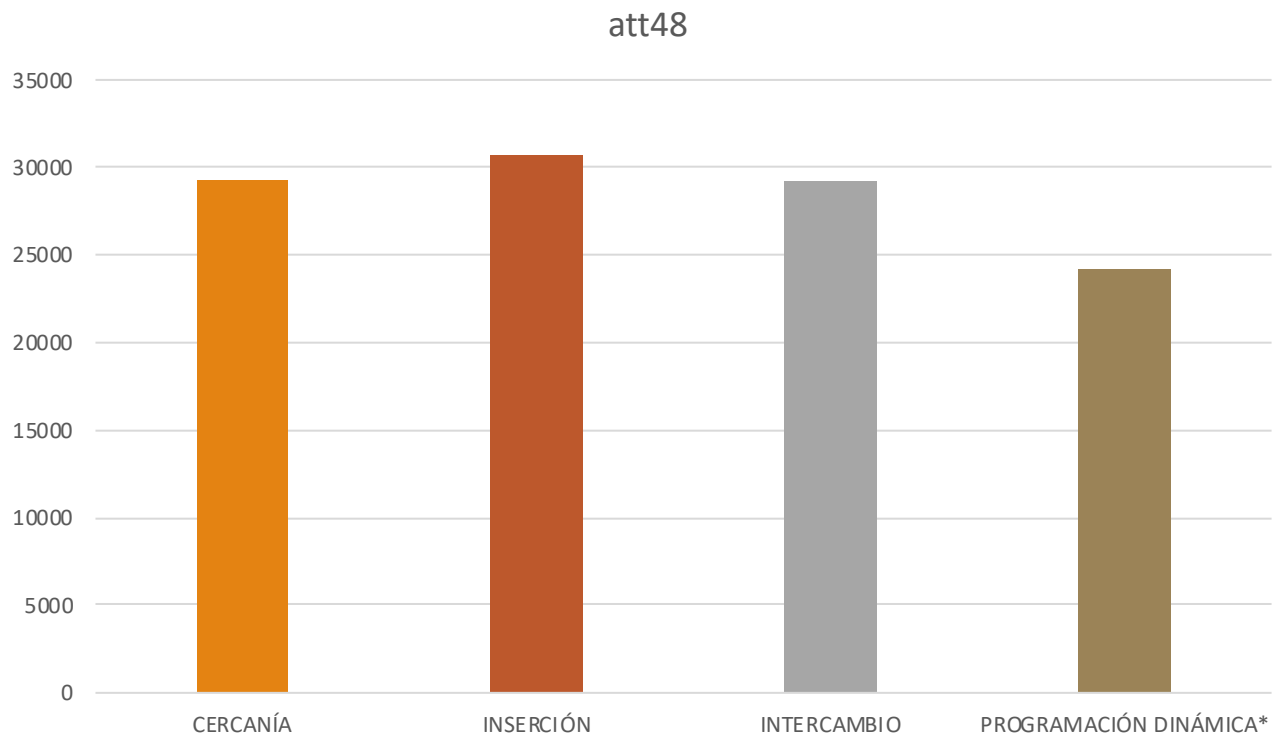




*Resultados obtenidos en computador con i7-8700B

VIAJANTE DE COMERCIO. PD.

COMPARACIÓN COSTES OBTENIDOS. GRÁFICAS



*Resultados obtenidos en computador con i7-8700B

CONCLUSIÓN

- Para qué tipo de problemas usar Programación Dinámica
- Mejor solución obtenida
- Mayor tiempo computacional

Gracias