

## Práctica 4

### Algorítmica

#### Introducción

#### Representación de la solución

#### Diseño del algoritmo: Branch and Bound

#### Pseudocódigo

#### Diseño del algoritmo: Backtracking

#### Pseudocódigo

#### Comparación

# TSP

## Algorítmica

Universidad de Granada

1 de junio de 2016

# Índice

## Práctica 4

### Algorítmica

#### Introducción

#### Representación de la solución

#### Diseño del algoritmo: Branch and Bound

#### Pseudocódigo

#### Diseño del algoritmo: Backtracking

#### Pseudocódigo

#### Comparación

- 1 Introducción
- 2 Representación de la solución
- 3 Diseño del algoritmo: Branch and Bound
- 4 Pseudocódigo
- 5 Diseño del algoritmo: Backtracking
- 6 Pseudocódigo
- 7 Comparación

# Introducción

## Práctica 4

### Algorítmica

### Introducción

Representación  
de la solución

Diseño del  
algoritmo:  
Branch and  
Bound

Pseudocódigo

Diseño del  
algoritmo:  
Backtracking

Pseudocódigo

Comparación

- El objetivo de esta práctica es resolver el problema del TSP utilizando para ello un enfoque Branch and Bound y, alternativamente, otro con Backtracking y comparar ambos.

# Representación de la solución

## Práctica 4

### Algorítmica

#### Introducción

#### Representación de la solución

#### Diseño del algoritmo: Branch and Bound

#### Pseudocódigo

#### Diseño del algoritmo: Backtracking

#### Pseudocódigo

#### Comparación

Vector de tamaño  $N$  en el que el índice del vector nos indica el orden en el que las ciudades deben de ser visitadas y el contenido de cada posición hace referencia al índice de la ciudad

# Diseño del algoritmo

## Práctica 4

### Algorítmica

#### Introducción

#### Representación de la solución

#### Diseño del algoritmo: Branch and Bound

#### Pseudocódigo

#### Diseño del algoritmo: Backtracking

#### Pseudocódigo

#### Comparación

Usaremos una cola con prioridad que será donde se introduzcan los nodos. En la cola se seleccionará el nodo con mayor prioridad, y de dicho nodo se consultará el valor de su cota local, que en caso de ser mejor a la global y ser nodo hoja, se actualizará la solución, en caso sólo de ser mejor que la global, se añadirán los hijos del nodo a la cola. En caso de ser peor, se devuelve la solución actual.

# Pseudocódigo

## Práctica 4

### Algorítmica

### Introducción

### Representación de la solución

### Diseño del algoritmo: Branch and Bound

### Pseudocódigo

### Diseño del algoritmo: Backtracking

### Pseudocódigo

### Comparación

```
Require: Matriz_costes, Vector_ciudades[N] Vector_distancias_minimas;  
Branch&Bound( Matriz_costes, Vector_ciudades  
Vector_distancias_minimas):  
priority_queue cola;  
solucion_final;  
Nodo n.generarnodo(Vector_ciudades[0])  
while !cola.empty() do  
    nodo = cola.top();  
    if EsHoja(nodo) && (nodo.cotalocal < cota global ) then  
        solucion_final = nodo.solucion;  
        cota global = nodo.cotalocal;  
    end if  
    if (nodo.cotalocal < cota global ) then  
        cola.add(nodo.generarhijos())  
        cola.push;  
    else  
        return solucion_final;  
    end if  
end while
```

# Diseño del algoritmo

## Práctica 4

### Algorítmica

#### Introducción

#### Representación de la solución

#### Diseño del algoritmo: Branch and Bound

#### Pseudocódigo

#### Diseño del algoritmo: Backtracking

#### Pseudocódigo

#### Comparación

La solución planteada es el mismo concepto que la aplicada en la anterior práctica, recorreremos el árbol en profundidad, calculando en cada nodo el valor de la cota local salvo que en caso de ser mayor que la global, podamos (ignoramos) a ese nodo y a todos sus hijos.

# Pseudocódigo

Práctica 4

Algorítmica

Introducción

Representación  
de la solución

Diseño del  
algoritmo:  
Branch and  
Bound

Pseudocódigo

Diseño del  
algoritmo:  
Backtracking

Pseudocódigo

Comparación

**Require:** Matriz, S\_final[N] S\_parcial[N] Ciudades[N]=false ciudad\_actual,  
nivel,valor\_maximo=0;

**Backtrack(S,S\_parcial,Ciudades,ciudad\_actual,nivel):**

Ciudades[ciudad\_actual]=true;

S\_parcial[nivel - 1]=ciudad\_actual;

**for** i to N **do**

**if** Ciudades[i]==false **then**

        valor\_actual = CalcularSolucionActual(S\_parcial);

**if** (cotalocal j cotaglobal) **then**

**Backtrack(S,S\_parcial,Ciudades,i,nivel+1);**

**end if**

**if** nodo\_actual == nodo\_hoja **then**

        valor\_actual = CalcularSolucionActual(S\_parcial)

**if** valor\_actual **mayor que** valor\_maximo **then**

            S\_final = S\_Actual

            valor\_maximo = valor\_actual

**end if**

**end if**

    Ciudades[i] = false;

**end if**



# Comparación

## Práctica 4

### Algorítmica

#### Introducción

#### Representación de la solución

#### Diseño del algoritmo: Branch and Bound

#### Pseudocódigo

#### Diseño del algoritmo: Backtracking

#### Pseudocódigo

#### Comparación

En ambos programas, hemos analizado los mismos mapas.  
Compararemos los tiempos y el empleo de los nodos para cada  
tipo de técnica empleada

# Comparación de tiempos

## Práctica 4

### Algorítmica

#### Introducción

#### Representación de la solución

#### Diseño del algoritmo: Branch and Bound

#### Pseudocódigo

#### Diseño del algoritmo: Backtracking

#### Pseudocódigo

#### Comparación

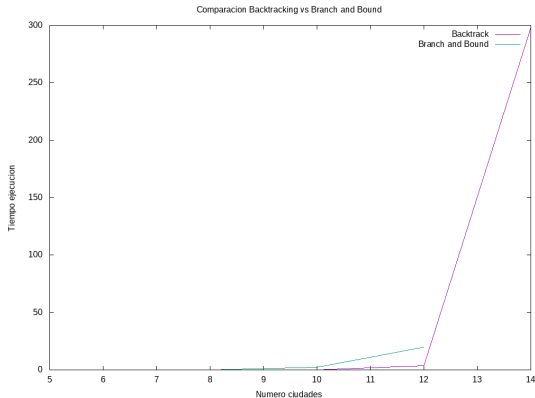


Figura: comparacion entre ambos

# Comparación de nodos: ulysses5

## Práctica 4

### Algorítmica

#### Introducción

#### Representación de la solución

#### Diseño del algoritmo: Branch and Bound

#### Pseudocódigo

#### Diseño del algoritmo: Backtracking

#### Pseudocódigo

#### Comparación

Técnica	Branch and bound	Backtracking
N.totales	120	120
N.podados	35	68
N.explorados	85	52

# Comparación de nodos: ulysses8

## Práctica 4

### Algorítmica

#### Introducción

#### Representación de la solución

#### Diseño del algoritmo: Branch and Bound

#### Pseudocódigo

#### Diseño del algoritmo: Backtracking

#### Pseudocódigo

#### Comparación

Técnica	Branch and bound	Backtracking
N.totales	40320	40320
N.podados	38568	37605
N.explorados	1752	2715

# Comparación de nodos: ulysses10

## Práctica 4

### Algorítmica

#### Introducción

#### Representación de la solución

#### Diseño del algoritmo: Branch and Bound

#### Pseudocódigo

#### Diseño del algoritmo: Backtracking

#### Pseudocódigo

#### Comparación

Técnica	Branch and bound	Backtracking
N.totales	3628800	3628800
N.podados	3601453	3566077
N.explorados	27347	62723

# Comparación de nodos: ulysses12

## Práctica 4

### Algorítmica

#### Introducción

#### Representación de la solución

#### Diseño del algoritmo: Branch and Bound

#### Pseudocódigo

#### Diseño del algoritmo: Backtracking

#### Pseudocódigo

#### Comparación

Técnica	Branch and bound	Backtracking
N.totales	479001600	479001600
N.podados	478284703	474474230
N.explorados	716897	4527370