

## Practica 3

### Algorítmica

#### Introducción

#### Primera versión

Diseño del  
algoritmo  
Pseudocódigo

#### Segunda versión

Diseño del  
algoritmo  
Pseudocódigo

#### Tercera versión

Diseño del  
algoritmo  
Pseudocódigo

#### Comparación

# Greedy: Travelling Salesman Problem

## Algorítmica

Universidad de Granada

3 de mayo de 2016

## Practica 3

### Algorítmica

#### Introducción

#### Primera versión

Diseño del algoritmo  
Pseudocódigo

#### Segunda versión

Diseño del algoritmo  
Pseudocódigo

#### Tercera versión

Diseño del algoritmo  
Pseudocódigo

#### Comparación

- 1 Introducción
- 2 Primera versión
  - Diseño del algoritmo
  - Pseudocódigo
- 3 Segunda versión
  - Diseño del algoritmo
  - Pseudocódigo
- 4 Tercera versión
  - Diseño del algoritmo
  - Pseudocódigo
- 5 Comparación

# Introducción

## Practica 3

### Algorítmica

### Introducción

#### Primera versión

Diseño del  
algoritmo  
Pseudocódigo

#### Segunda versión

Diseño del  
algoritmo  
Pseudocódigo

#### Tercera versión

Diseño del  
algoritmo  
Pseudocódigo

#### Comparación

- El objetivo de esta práctica es diseñar varios algoritmos Greedy resolviendo el problema del viajante de comercio o TSP

# Primera versión

## Practica 3

### Algorítmica

#### Introducción

#### Primera versión

Diseño del algoritmo  
Pseudocódigo

#### Segunda versión

Diseño del algoritmo  
Pseudocódigo

#### Tercera versión

Diseño del algoritmo  
Pseudocódigo

#### Comparación

Aplicar algoritmo usando Greedy simple, buscaremos en cada momento la ciudad más cercana y esa ciudad la añadiremos a nuestro conjunto de soluciones.

# Diseño del algoritmo

## Practica 3

### Algorítmica

#### Introducción

#### Primera versión

#### Diseño del algoritmo Pseudocódigo

#### Segunda versión

#### Diseño del algoritmo Pseudocódigo

#### Tercera versión

#### Diseño del algoritmo Pseudocódigo

#### Comparación

- **Conjunto de candidatos:** Conjunto de ciudades. (Conjunto **C**)
- **Conjunto de seleccionados:** Conjunto de ciudades ordenado. (Conjunto **S**)
- **Función solución:** Cuando el conjunto de candidatos esté vacío.
- **Función factibilidad:** En principio no hay función de factibilidad.
- **Función selección:** Se seleccionará el día más cercano no visitado.
- **Función objetivo:** Lista con las ciudades en el orden en el que hay que visitarlas.

# Pseudocódigo I

## Practica 3

### Algorítmica

#### Introducción

#### Primera versión

#### Diseño del algoritmo Pseudocódigo

#### Segunda versión

#### Diseño del algoritmo Pseudocódigo

#### Tercera versión

#### Diseño del algoritmo Pseudocódigo

#### Comparación

## Selección

**Require:** Conjunto de ciudades C,S

x=0

S=C[0]

**for** i = 1 to len(C) **do**

**if** C[i] MenorDistancia **then**

        x = C[i]

        S.add(x)

        C[i].erase

**end if**

**end for**

**return** S

# Segunda versión

## Practica 3

### Algorítmica

#### Introducción

#### Primera versión

Diseño del  
algoritmo  
Pseudocódigo

#### Segunda versión

Diseño del  
algoritmo  
Pseudocódigo

#### Tercera versión

Diseño del  
algoritmo  
Pseudocódigo

#### Comparación

Aplicar algoritmo usando Greedy con selección parcial, estableceremos un recorrido inicial seleccionando las 3 ciudades más alejadas (oeste,norte y este). A partir de ahí añadiremos las ciudades con respecto a los intervalos creados.

# Diseño del algoritmo

## Practica 3

### Algorítmica

#### Introducción

#### Primera versión

Diseño del algoritmo  
Pseudocódigo

#### Segunda versión

Diseño del algoritmo  
Pseudocódigo

#### Tercera versión

Diseño del algoritmo  
Pseudocódigo

#### Comparación

- **Conjunto de candidatos:** Conjunto de ciudades. (Conjunto **C**)
- **Conjunto de seleccionados:** Conjunto de ciudades ordenado. (Conjunto **S**)
- **Función solución:** Cuando el conjunto de candidatos esté vacío.
- **Función factibilidad:** En principio no hay función de factibilidad.
- **Función selección:** Se seleccionará una ciudad y comprobaremos para cada una de las posiciones del circuito solución mitocondria el mejor lugar para insertarlo (la que ofrece menor distancia) .
- **Función objetivo:** Lista con las ciudades en el orden en el que hay que visitarlas.



# Pseudocódigo I

## Practica 3

### Algorítmica

#### Introducción

#### Primera versión

#### Diseño del algoritmo Pseudocódigo

#### Segunda versión

#### Diseño del algoritmo Pseudocódigo

#### Tercera versión

#### Diseño del algoritmo Pseudocódigo

#### Comparación

**Require:** Conjunto de ciudades C,S

x=0

S=CalcularRecorridoInicial()

**for** i=0 to len(C) **do**

    x = C[i]

**for** j=0 to len(S) **do**

        x = C[i]

        MejorPosicion(S[j],x)

        C[i].erase

**end for**

**end for**

**return** S

# Tercera versión

## Practica 3

### Algorítmica

#### Introducción

#### Primera versión

Diseño del algoritmo  
Pseudocódigo

#### Segunda versión

Diseño del algoritmo  
Pseudocódigo

#### Tercera versión

Diseño del algoritmo  
Pseudocódigo

#### Comparación

Implementación del algoritmo Greedy 2-Opt. Calcularemos una primera aproximación de la solución utilizando el algoritmo de la primera versión. Después, iremos para cada ciudad intercambiando con las siguientes, viendo si alguno de esos intercambios mejora la primera solución obtenida, así hasta intentar generar una solución mejor que la primera

# Diseño del algoritmo

## Practica 3

### Algorítmica

### Introducción

### Primera versión

### Diseño del algoritmo Pseudocódigo

### Segunda versión

### Diseño del algoritmo Pseudocódigo

### Tercera versión

### Diseño del algoritmo Pseudocódigo

### Comparación

- **Conjunto de candidatos:** Conjunto de ciudades. (Conjunto **C**)
- **Conjunto de seleccionados:** Conjunto de ciudades ordenado. (Conjunto **S**)
- **Función solución:** Cuando el conjunto de candidatos esté vacío.
- **Función factibilidad:** En principio no hay función de factibilidad.
- **Función selección:** Dada la solución con la primera versión del algoritmo, comprobaremos si intercambiando una por una las ciudades se mejora la solución inicial .
- **Función objetivo:** Lista con las ciudades en el orden en el que hay que visitarlas.

# Pseudocódigo I

Practica 3

Algorítmica

Introducción

Primera  
versión

Diseño del  
algoritmo  
Pseudocódigo

Segunda  
versión

Diseño del  
algoritmo  
Pseudocódigo

Tercera  
versión

Diseño del  
algoritmo  
Pseudocódigo

Comparación

**Require:** Conjunto de ciudades C,S,AUX

S=PrimeraVersión();Z=Distancia(S);

x,y,i = 0; j = 1;

**while** i menor o igual **tam - 2** and **j** menor o igual a **tam - 1**  
**do**

Sp=Swap(S,i,j);Zp=Distancia(Sp);

**if** Zp mayor o igual que z and **j** menor que **tam-1** **then**  
j++;

**end if**

**if** zp mayor o igual que z and **j** == **tam - 1** **then**  
j += 1; i += 1

**end if**

**if** zp menor que z **then**

S = Sp; Z = Zp;

i += i; j = i + 1;

**end if**

# Comparación recorridos

## Practica 3

### Algorítmica

#### Introducción

#### Primera versión

Diseño del  
algoritmo  
Pseudocódigo

#### Segunda versión

Diseño del  
algoritmo  
Pseudocódigo

#### Tercera versión

Diseño del  
algoritmo  
Pseudocódigo

#### Comparación

En esta sección veremos los distintos resultados obtenidos de cada una de las ejecuciones de los algoritmos.

Mapa	Versión I	Versión II	Versión III	Versión óptima
Berlín52	8314.81	9622.17	7883.92	7542
Ch130	7022.12	7499.39	6738.84	6110
Gr666	25345.7	27646.1	24163.5	21282

# Comparación gráficas

## Practica 3

### Algorítmica

#### Introducción

#### Primera versión

Diseño del algoritmo  
Pseudocódigo

#### Segunda versión

Diseño del algoritmo  
Pseudocódigo

#### Tercera versión

Diseño del algoritmo  
Pseudocódigo

#### Comparación

En esta sección veremos los distintos grafos obtenidos de cada uno de los algoritmos para el mapa Berlin52

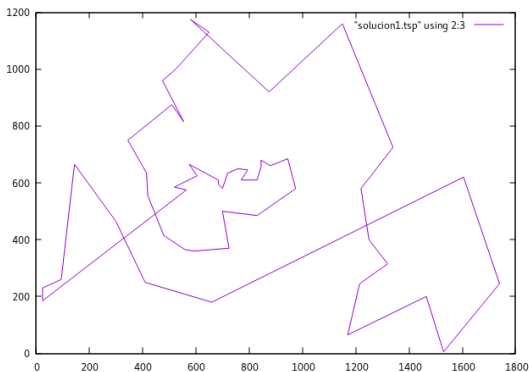


Figura: Primera versión

# Comparación gráficas

## Practica 3

### Algorítmica

#### Introducción

#### Primera versión

Diseño del algoritmo  
Pseudocódigo

#### Segunda versión

Diseño del algoritmo  
Pseudocódigo

#### Tercera versión

Diseño del algoritmo  
Pseudocódigo

#### Comparación

En esta sección veremos los distintos grafos obtenidos de cada uno de los algoritmos para el mapa Berlin52

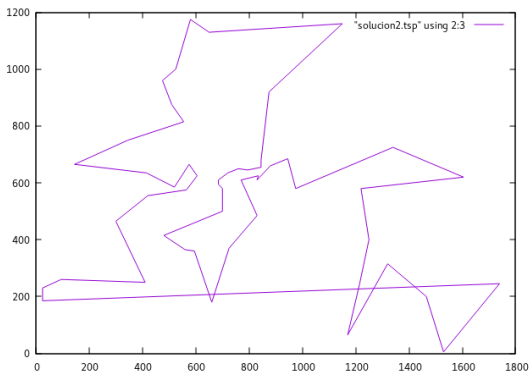


Figura: Segunda versión

# Comparación gráficas

## Practica 3

### Algorítmica

#### Introducción

#### Primera versión

Diseño del algoritmo  
Pseudocódigo

#### Segunda versión

Diseño del algoritmo  
Pseudocódigo

#### Tercera versión

Diseño del algoritmo  
Pseudocódigo

#### Comparación

En esta sección veremos los distintos grafos obtenidos de cada uno de los algoritmos para el mapa Berlin52

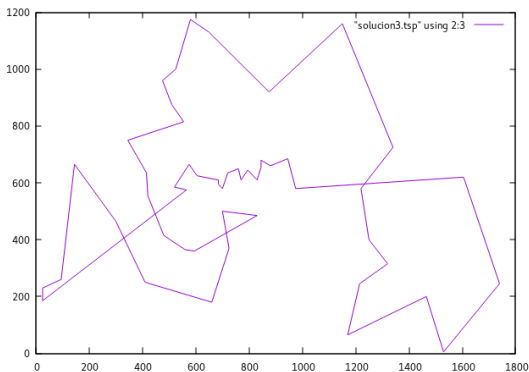


Figura: Tercera versión