

# Problema del viajante de comercio

## Travelling salesman problem

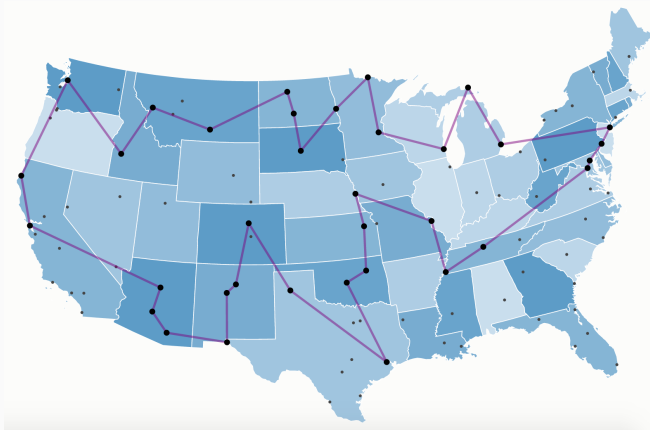
---

José Antonio Álvarez Ocete  
Norberto Fernández de la Higuera  
Javier Gálvez Obispo  
Yábir García Benchakhtir

2 de mayo de 2018

Doble Grado en Ingeniería Informática y Matemáticas

# Descripción del problema



---

**Algorithm 1** Nearest Neighbor

---

```
 $\mathcal{R} = [\text{random}(0, |\Omega|)]$   
 $\mathcal{C} \leftarrow \mathcal{R}[0]$   
for pos in  $[0, n-2]$  do  
    bestPos = nearest( $\mathcal{M}[\mathcal{R}[\text{pos}]]$ ,  $\mathcal{C}$ )  
     $\mathcal{R}.\text{push}(\mathcal{C}[\text{bestPos}])$   
     $\mathcal{C}.\text{erase}(\text{bestPos})$   
end for  
return  $\mathcal{R}$ 
```

---

# Ejemplo de ejecución

1 2

---

<sup>1</sup>Si no se muestra correctamente se adjunta el ejemplo en el archivo nearest.gif

<sup>2</sup>También está disponible en <https://imgur.com/a/iyweYV6>

---

**Algorithm 2** Cheap Insert

---

$i, j = \text{nodes}(\min(M))$

$\text{path} = [i, j]$

$\mathcal{C} = \Omega - \text{path}$

**while**  $\mathcal{C}$  not empty **do**

**for** city in cities **do**

**if** city not in path **then**

**for** node in path **do**

                minimizeDistance(node, city, node+1)

**end for**

            insert(path, T)

            remove( $\mathcal{C}$ , T)

**end if**

**end for**

**end while**

# Ejemplo de ejecución

1 2

---

<sup>1</sup>Si no se muestra correctamente se adjunta el ejemplo en el archivo nearest.gif

<sup>2</sup>También está disponible en <https://imgur.com/a/iyweYV6>