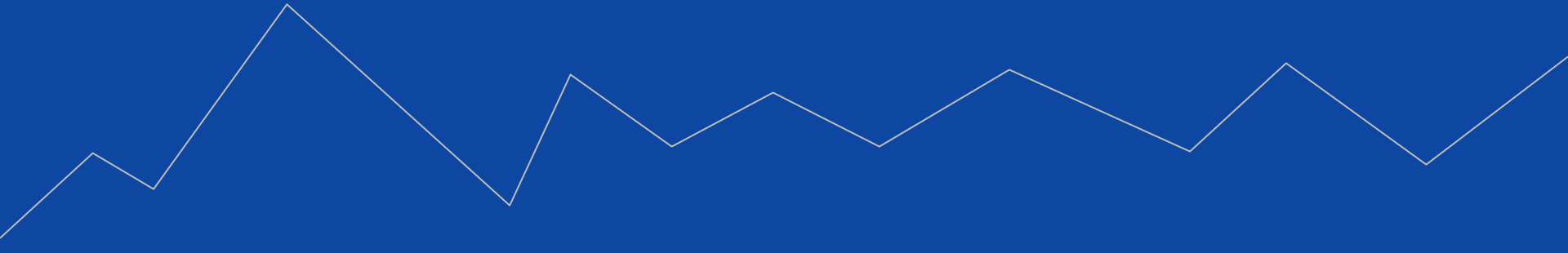


# Heurísticas

Algoritmos y Estructuras de Datos 3



# Qué es una heurística?

Un algoritmo heurístico, también llamado heurística es un algoritmo que no garantiza una respuesta exacta para el problema en cuestión.

# Qué es una heurística?

Un algoritmo heurístico, también llamado heurística es un algoritmo que no garantiza una respuesta exacta para el problema en cuestión.

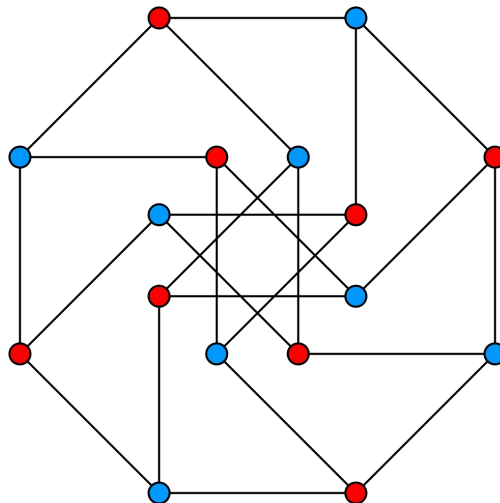
Hay problemas que son difíciles...



# Qué es una heurística?

Un algoritmo heurístico, también llamado heurística es un algoritmo que no garantiza una respuesta exacta para el problema en cuestión.

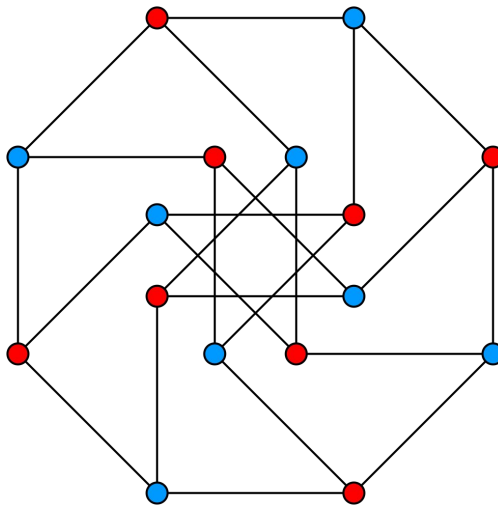
Hay problemas que son difíciles...



# Qué es una heurística?

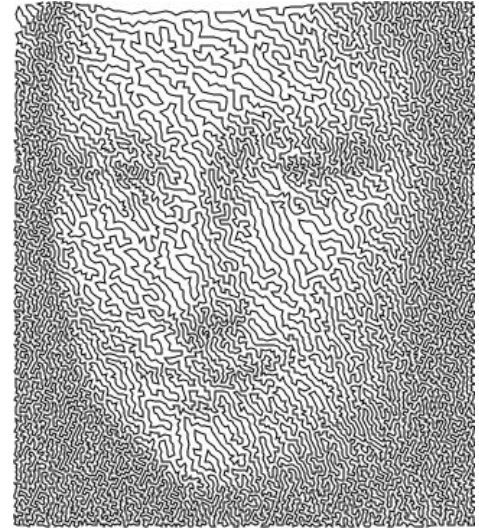
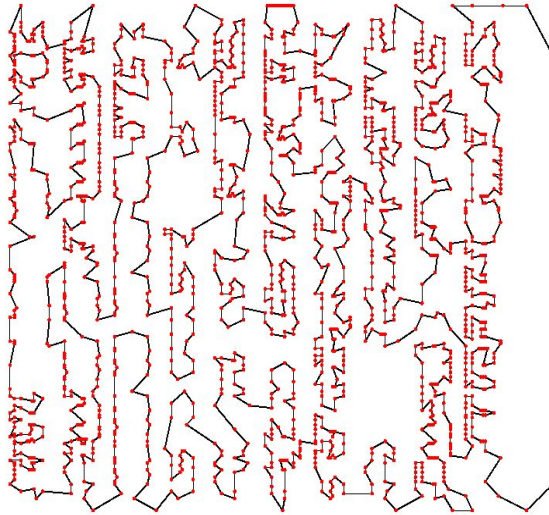
Un algoritmo heurístico, también llamado heurística es un algoritmo que no garantiza una respuesta exacta para el problema en cuestión.

Hay problemas que son difíciles...



# Problema del Viajante de Comercio

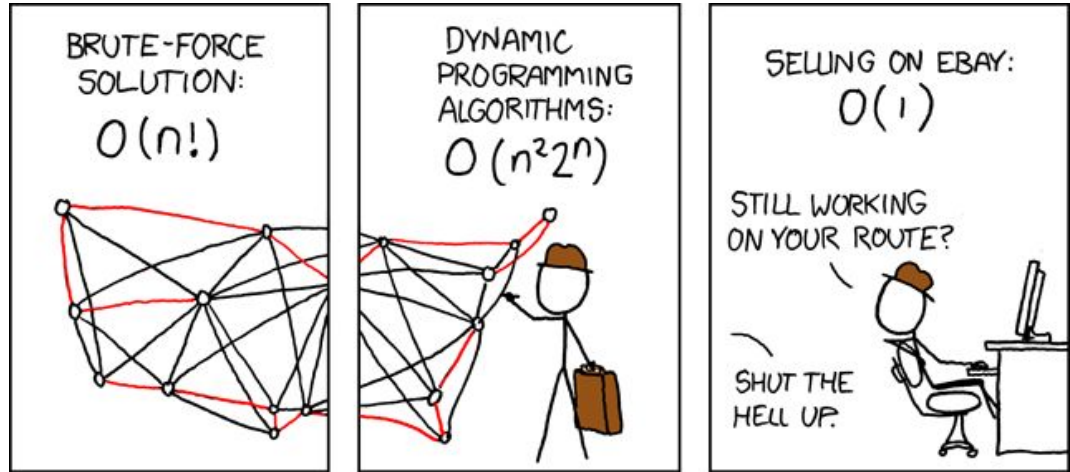
- Conocido por sus abreviaciones en inglés TSP (travelling salesman problem)
- Es tan importante que tiene película propia<sup>1</sup>.
- Existen muchas variantes a este problema, por ejemplo, nuestro TP.



[1] [www.imdb.com/title/tt1801123/](http://www.imdb.com/title/tt1801123/)

# TSP es difícil

- Los mejores algoritmos resuelven menos de 90k nodos.
- Problema clásico para probar heurísticas.
- En 1954 Dantzig, Fulkerson y Johnson resuelven una instancia de 49 nodos.
- Hace poco salió una app, Google Trips, que utiliza una heurística de TSP para recorrer puntos interesante en una ciudad<sup>1</sup>.

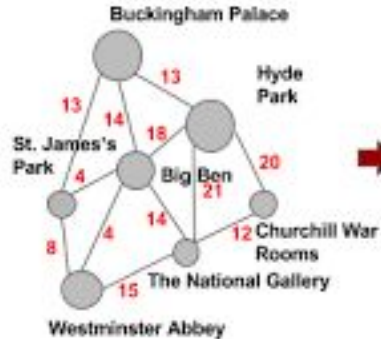


[1] [research.googleblog.com/2016/09/the-280-year-old-algorithm-inside.html](https://research.googleblog.com/2016/09/the-280-year-old-algorithm-inside.html)

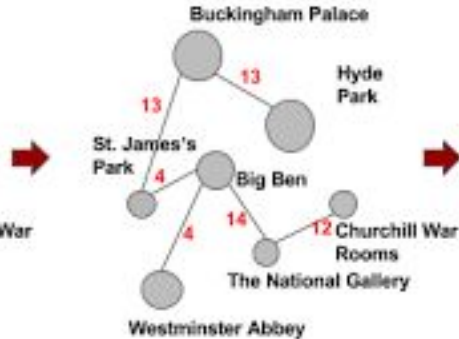


# Heurísticas Greedy

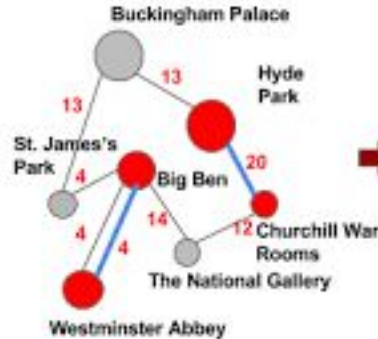
- Vecino más cercano
- Vecino más lejano
- Vecino más barato
- Christofides



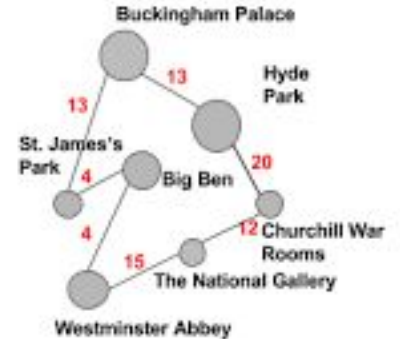
Begin with the location graph.



Construct a MST over all vertices.



Compute a min-cost matching to ensure every vertex has an even degree.



Compute Eulerian tour.

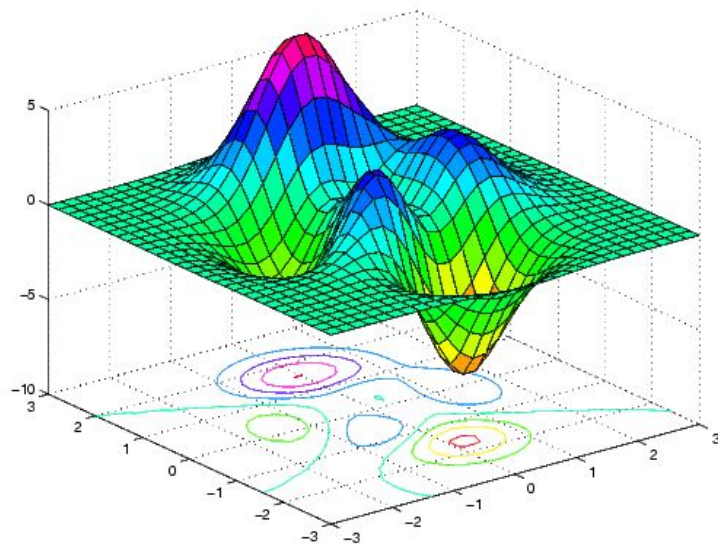


# Búsqueda local

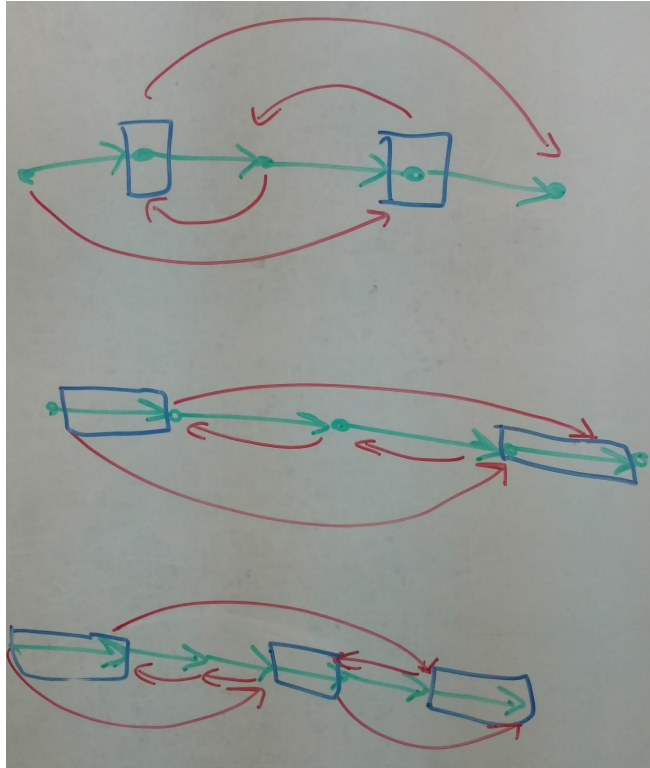
La idea es definir un vecindario de soluciones. Depende de nuestra representación y estructuras, algunas vecindades son más apropiadas que otras.

El tamaño del vecindario implica un *tradeoff* entre performance computacional y calidad de la solución.

Para el TSP existen varios vecindarios posibles, los más comunes son intercambiar ejes o nodos.



# Algunos vecindarios TSP



- SWAP - Intercambia la posición de dos vertices. Facil de implementar.
- 2OPT - Intercambia posición de aristas. Costo nuevo fácil de calcular.
- 3OPT - Extensión de 2OPT

# Metaheurísticas

Heurísticas de alto nivel. Se usan para escapar de mínimos locales.

No necesariamente exploran todo el vecindario de una solución.

En algunos casos vuelven a comenzar desde una solución inicial distinta.

En general, las condiciones de terminación cumplen alguna condición.

Tabu, GRASP, Simulated Annealing, Algoritmos Genéticos, POPMUSIC, Colonia de hormigas, etc etc etc.

# Ejercicio

Utilizar el archivo provisto por la materia para encontrar soluciones del problema del viajante de comercio.

- Modificar generarSolucion con una heurística constructiva.
- Modificar mejorarSolucion con una búsqueda local.

Al final de la clase vemos quien llega a mejores resultados con la constructiva y con la búsqueda local para distintas semillas.