

# TSP - PROGRAMACION II

## ESTRATEGIAS DE RESOLUCIÓN DE PROBLEMAS

## Algoritmos de Recursión Simple

- Básicamente consiste en algoritmos en donde aplicacion recursión para resolver problemas simples.
- Pueden ser simples implementaciones de métodos iterativos expresados como métodos recursivos.
- Se debe analizar el problema y en caso de poder escribirlo como iterativo, se debe optar siempre por este camino, debido a las desventajas de los algoritmos recursivos:
  - Lentos
  - Consumo de memoria

## Divide y Vencerás

- Divide el problema de tamaño  $N$  en problemas más pequeños, cada uno de los cuales es similar al original pero de menor tamaño.
  - Si se llega a una solución de los subproblemas, se podrá construir de forma más sencilla una solución al problema general.
  - Cada uno de estos subproblemas se podrá resolver de forma directa (caso base) o dividiéndolos en problemas más pequeños mediante la recursividad.
  - Los ejemplos que se han hecho utilizan la estrategia de divide y vencerás.
- Ejemplos:
  - Encontrar el factorial de un número.
  - La serie de Fibonacci.

## Backtracking

- Divide la solución en pasos, en cada uno de los cuales hay una serie de opciones que ha de probar de manera sistemática.
- En cada paso se busca una posibilidad o solución aceptable.
  - Si se encuentra se pasa a decidir el siguiente paso.
  - Si no se encuentra una solución aceptable, se retrocede hasta la última solución aceptable encontrada y se elige una opción distinta a la anterior.
- La **recursividad** se utiliza para poder retroceder hasta encontrar una solución aceptable.
- Ejemplos:
  - Laberinto
  - Juegos de tablero
  - etc.

## Programación Dinámica

- La base de la programación dinámica es el razonamiento inductivo:
  - ¿cómo resolver un problema combinando soluciones para problemas más pequeños?
- La idea es la misma que en **divide y vencerás**, pero aplicando una estrategia distinta.

### Similitud:

- Descomposición **recursiva** del problema.
- Se obtiene aplicando un razonamiento inductivo.

### Diferencia:

- Divide y vencerás: aplicar directamente la fórmula recursiva (programa recursivo).
- Programación dinámica: resolver primero los problemas más pequeños, guardando los resultados en una tabla (programa iterativo).

## Algoritmos Golosos (Greedy)

- Un algoritmo goloso es un algoritmo que va encontrando óptimos locales para subproblemas para en base a los óptimos locales encontrar un óptimo global que contiene a los óptimos locales.
- Diremos que un algoritmo es greedy cuando en cada paso, elige la 'mejor' solución local.
- Desarrollar un algoritmo recursivo, pero NO se usa la recurrencia para el algoritmo, se basa en una elección en cada paso.

## Optimización de Problemas

La optimización de problemas consiste no sólo en encontrar la solución al problema, sino en encontrar la MEJOR solución al problema.

Estrategias para buscar optimizaciones de problemas:

- Programación Dinámica
- Algoritmos Golosos (Greedy)