

## Número de caminos

### Objetivos

- Diseñar algoritmos usando las estrategias de diseño de algoritmos vistas en el temario para resolver un problema de la forma más eficiente posible.
- Saber explicar el funcionamiento de un algoritmo de forma que sea entendible por una persona "ajena" a la asignatura.
- Saber escribir el pseudocódigo de un algoritmo.
- Implementar un algoritmo en C++ a partir de su pseudocódigo y verificar su correcto funcionamiento.
- Calcular la complejidad temporal asintótica de un algoritmo.
- Manejar la bibliografía fundamental de la materia, así como las diferentes fuentes y recursos on-line.

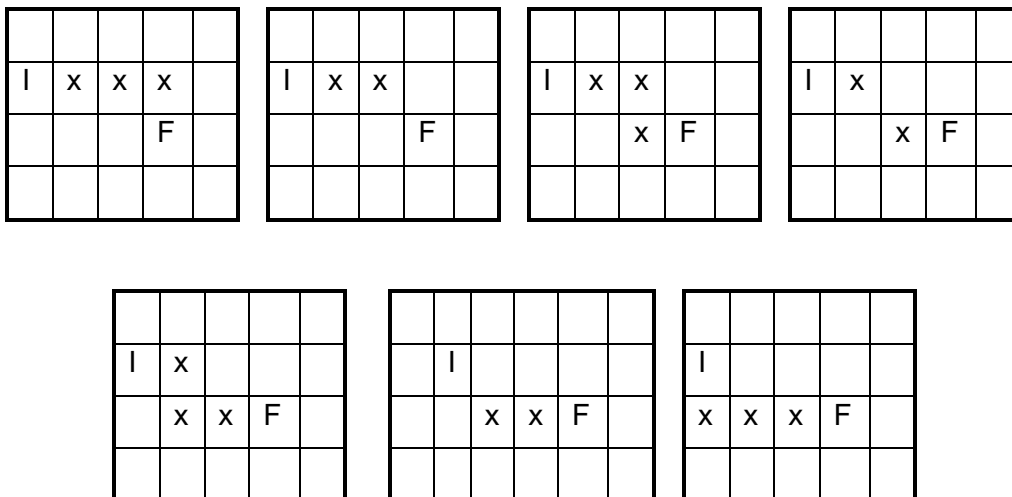
### Enunciado

Se tiene un robot que puede realizar tres tipos de movimientos: derecha, abajo y en diagonal hacia la derecha y abajo con un ángulo de  $315^\circ$  y un plano con X filas e Y columnas. Se coloca el robot en un posición del plano indicada por su fila y columna y se quiere obtener un algoritmo que calcule el número total de caminos posibles que puede realizar el robot desde dicho punto hasta un punto final.

#### Ejemplo:

Si se dispone de un plano de tamaño 4 filas y 5 columnas y el robot se encuentra en el punto inicial  $I=(2,1)$  y el punto final es  $F=(3,4)$ , el robot puede realizar 7 posibles caminos.

Estos caminos son los siguientes:



### Actividades

#### Actividad 1: Memoria

Se realizará una memoria que contendrá:

- **PORTADA:**

Una hoja con el nombre de la asignatura, el nombre de la práctica ("Número de caminos"), el curso académico (2022/2023) y los nombres y apellidos de los componentes del grupo.

- **ÍNDICE O TABLA DE CONTENIDO:**

La siguiente página de la memoria contendrá el índice del trabajo. En el índice se incluirán los apartados descritos a continuación **manteniendo los nombres y la numeración**, indicando en la tabla de contenido el número de página donde comienza cada apartado/subapartado. Los apartados se pueden dividir en los subapartados que se quieran.

## **1.- Algoritmo recursivo**

### **1.1.- Descripción del algoritmo.**

Descripción **detallada** del funcionamiento de un algoritmo recursivo que resuelva el problema y que se ajuste exclusivamente a lo que se pide en el enunciado. Esta descripción debe realizarse de forma que quede claro su funcionamiento y sea entendible por una persona ajena a la asignatura (por ejemplo, amistades que no estudien informática, hermanos/-as, padres,...).

### **1.2.- Estrategia de programación y tipo de recursividad del algoritmo.**

Se debe justificar adecuadamente el tipo de recursividad que tiene el algoritmo.

### **1.3.- Pseudocódigo.**

Pseudocódigo del algoritmo utilizando la nomenclatura vista en la asignatura.

### **1.4. Ejemplo.**

Inclusión de un ejemplo de aplicación a unos puntos del plano y realización de una traza detallada del algoritmo. El ejemplo será distinto al que se incluye en el enunciado de esta práctica.

### **1.5.- Complejidad asintótica.**

Análisis y justificación de la complejidad asintótica del algoritmo en pseudocódigo.

## **2.- Algoritmo iterativo.**

### **2.1.- Descripción.**

Descripción **detallada** del funcionamiento de un algoritmo iterativo que resuelva el problema lo más eficientemente posible y que se ajuste exclusivamente a lo que se pide en el enunciado. Esta descripción debe realizarse de forma que quede claro su funcionamiento y sea entendible por una persona ajena a la asignatura (por ejemplo, amistades que no estudien informática, hermanos/-as, padres...).

### **2.2.- Estrategia de programación.**

Identificación y justificación de la estrategia general de programación que se aplica (Divide y vencerás, Voraz, Programación Dinámica, Vuelta atrás, Ramificación y poda).

### **2.3.- Pseudocódigo.**

Pseudocódigo del algoritmo utilizando la nomenclatura vista en la asignatura.

### **2.4. Ejemplo.**

Inclusión de un ejemplo de aplicación a unos puntos del plano y traza detallada del algoritmo. El ejemplo será distinto al que se incluye en el enunciado de esta práctica.

### **2.5.- Complejidad asintótica.**

Análisis y justificación de la complejidad asintótica del algoritmo en pseudocódigo.

### 3.- Bibliografía

Especificación toda la bibliografía consultada con el siguiente formato:

- Para libros, apuntes,...:  
Título:  
Autor/-es:  
Año:
- Para direcciones web:  
Título:  
Url: (enlace **directo** a la página web)

El formato para la memoria es:

- Títulos de los apartados: Arial, 12
- Texto general: Arial, 10
- Interlineado: Sencillo o simple.
- Párrafo: Espaciado anterior y posterior: 0
- Márgenes: 2 cm. a cada lado.
- Las páginas irán numeradas y la numeración se corresponderá con la numeración de la tabla de contenido que se indica en el índice.

### Actividad 2: Programa

Implementación de un programa en C/C++ correspondiente al pseudocódigo de los algoritmos de la memoria (siguiendo el modo de transformación visto en la asignatura). El programa presentará un menú con las siguientes opciones:

Plano: X x Y  
Punto inicial: (x1, y1)  
Punto final: (x2, y2)

- 1.- Tamaño del plano.
- 2.- Puntos inicial y final.
- 3.- Algoritmo recursivo.
- 4.- Algoritmo iterativo.
- 5.- Salir.

donde X, Y, x1, y1, x2 e y2 son el número de filas y columnas del plano, del punto inicial y final, respectivamente. El valor inicial de todos estos valores es 0 y se irán actualizando según se haya asignado su valor en las correspondientes opciones del menú

A continuación, se detalla qué debe realizar cada opción.

#### 1.- Tamaño del plano.

Se pedirá el número de filas y columnas del plano.

#### 2.- Puntos inicial y final.

Se pedirán los números de fila y columna de los puntos inicial y final.

#### 3.- Algoritmo recursivo.

#### 4.- Algoritmo iterativo.

Se obtendrá el número total de caminos para ir del punto inicial al final aplicando el algoritmo correspondiente.

#### 5.- Salir.

Con esta opción se saldrá del programa. Mientras no se pulse esta opción se volverá a presentar el menú del programa para elegir cualquiera de las opciones del mismo.

## Modo de entrega

La práctica se realizará **en grupos de 2 o 3 componentes**.

Uno de los componentes del grupo entregará un archivo comprimido llamado **ncaminos.zip** en la tarea correspondiente al proyecto de la Convocatoria Ordinaria (Evaluación Continua) dentro del campus virtual de la asignatura. Este archivo **solamente** contendrá los siguientes ficheros: **ncaminos.pdf** (memoria de la práctica), **ncaminos.cpp** (código fuente del programa) y **ncaminos.exe** (ejecutable). Indicad los nombres de los miembros del grupo tanto en el fichero .pdf como en el .cpp.

**Fecha fin de entrega:** 15 de junio de 2023 a las 23:59.

## Evaluación

Esta práctica constituye 2 puntos de la nota final. La distribución de la puntuación es la siguiente:

Algoritmo recursivo: 1,2 puntos

Algoritmo iterativo: 0,8 puntos

La realización de un algoritmo comprende el desarrollo de la memoria y del programa.

Se valorará:

- Memoria
  - Desarrollo correcto de los apartados especificados.
  - Algoritmos propuesto para la resolución del problema. Se valora que sean algoritmos lo más eficiente posible.
  - Pseudocódigo según la notación vista de clase.
  - Claridad en la redacción, sin errores de sintaxis (faltas de ortografía, tildes,...)
- Programa
  - El correcto funcionamiento del programa.
  - El formato de salida por pantalla, claridad y facilidad de manejo del programa.
  - Adecuación de la implementación del algoritmo con el pseudocódigo de la memoria.
- Entrevista personal
  - Se podrá requerir la presencia individual de los componentes del grupo donde el profesor le realizará preguntas sobre la práctica, de forma que las respuestas del alumno serán decisivas para la nota final del alumno en la práctica, pudiendo haber diferentes puntuaciones entre los miembros de un grupo según la entrevista personal.
- **Solamente** se evaluarán las prácticas:
  - Entregadas según las especificaciones indicadas en el apartado “Modo de entrega”.
  - Sin errores de compilación utilizando los compiladores instalados en el aula de prácticas o máquinas virtuales. Estos compiladores han sido utilizados durante las sesiones de prácticas y son: Visual Studio Code y CodeBlocks.
  - Sin virus.

### IMPORTANTE:

La nota de la práctica será un 0 si:

- Falta algún apartado de la memoria de la opción entregada.
- Se utiliza pseudocódigo que no se corresponde con el visto en la asignatura.
- Falta el programa que implemente el algoritmo/-s descritos en la memoria.
- El programa implementado no se corresponde con el pseudocódigo descrito en la memoria.
- El código del programa da errores de compilación utilizando los compiladores descritos en el punto anterior.
- El código utiliza librerías no vistas en las prácticas de la asignatura sin solicitar y recibir su autorización por parte del profesorado.
- El programa no obtiene correctamente la solución del problema.

- Los nombres de los ficheros entregados no se ajustan a los nombres especificados en el apartado “Modo de entrega” y/o se incluyen más ficheros que los especificados.
- Se detectan indicios de copia entre prácticas. En este caso la nota de **TODAS las prácticas implicadas** será un 0. No se contempla ningún tipo de justificaciones: “Hemos comentado cosas de la práctica entre nosotros,...”, “la práctica la he hecho yo y él/ella se la ha copiado”, “no sé cómo ha llegado la práctica a manos de otra persona”,....)
- El alumno no defiende adecuadamente la práctica en la entrevista personal.

iii COMPROBAD ESTOS PUNTOS ANTES DE ENTREGAR LA PRÁCTICA !!!