

ALGORÍTMICA

Curso 2016/17

Datos Académicos

- ♦ Luis M. de Campos Ibáñez
- ♦ Dpto. Ciencias de la Computación e I.A.
- ♦ 4ª planta, despacho 10
- ♦ Email: ICI@decsai.ugr.es
- ♦ Tutorías:
 - ♦ Martes: 9:30 a 11:30 horas
 - ♦ Miércoles: 11:30 a 13:30 horas
 - ♦ Jueves: 9:30 a 10:30 y de 11:30 a 12:30 horas

Horario

2º A Grado en Informática 2º cuatrimestre															
	Lunes			Martes			Miércoles			Jueves			Viernes		
8:30-9:30															
9:30-10:30	AC (A2) 3.7	FIS (A3) 2.7	FBD (A1) 2.6	FIS (A1) 3.7	IA (A2) 2.7	FBD (A3) 2.6	FBD (A2) 2.6	FBD (A1) 3.7	IA (A3) 2.7	IA (A1) 2.7	FBD (A2) 2.6	AC (A3) 3.7	ALG (A3) 2.6	AC (A1) 3.7	FIS (A2) 2.7
10:30-11:30	AC (A2) 3.7	FIS (A3) 2.7	FBD (A1) 2.6	FIS (A1) 3.7	IA (A2) 2.7	FBD (A3) 2.6	FBD (A2) 2.6	ALG (A1) 3.7	IA (A3) 2.7	IA (A1) 2.7	ALG (A2) 2.6	AC (A3) 3.7	FBD (A3) 2.2	AC (A1) 3.7	FIS (A2) 2.7
11:30-12:30	IA 0.1			ALG 0.1			FIS 0.1			AC 0.1			FBD 0.1		
12:30-13:30	IA 0.1			ALG 0.1			FIS 0.1			ALG 0.1			AC 0.1		
13:30-14:30															
15:30-16:30															
16:30-17:30															
17:30-18:30															
18:30-19:30															
19:30-20:30															
20:30-21:30															

Plataforma Docente en Decsai

- ♦ **<http://decsai.ugr.es>**
- ♦ Los alumnos que hayan empleado la aplicación en el curso anterior accederán con el mismo login y contraseña que tuvieran
- ♦ Acceso por primer vez: Emplear como login y contraseña su D.N.I. o pasaporte.
 - ♦ El sistema hará uso de la cuenta de correo de la Universidad del alumno para enviarle un correo que le permita activar su cuenta, si bien los mensajes de la plataforma se enviarán a la cuenta de correo que indique como habitual

Datos Académicos: Prácticas

- ♦ Luis M. de Campos (A1, A2 y A3)
- ♦ Organización de las prácticas en 3 Subgrupos:
 - ♦ Cada estudiante debe escoger un subgrupo. El plazo para hacerlo es entre el 14 y el 21 de febrero, ambos inclusive (hasta las 15:00 h).
 - ♦ Puede hacerlo a través de la plataforma de docencia en decsai. El tamaño máximo por cada grupo de prácticas es de 29.

Organización de las prácticas

- ♦ Posteriormente los estudiantes de cada subgrupo formarán equipos, que serán **estables** durante todo el cuatrimestre.
- ♦ Tamaño típico de cada equipo: 5 miembros, pudiendo excepcionalmente ser de menos miembros (enviar por correo al profesor del subgrupo la composición del equipo).
- ♦ El plazo es hasta el 3 de marzo.
- ♦ Las prácticas empezarán la semana del 6 de marzo.

Desarrollo de las prácticas

Se realizarán prácticas semanales sobre resolución de problemas y desarrollo de proyectos relativos a los temas explicados en las clases de teoría

- Cada dos semanas se propondrá un conjunto de problemas o trabajos, que se sortearán entre los diferentes equipos.
- A la siguiente semana, cada equipo tendrá que exponer en clase de prácticas el problema o trabajo asignado.
- El miembro del equipo que exponga se elegirá al azar.
- La asistencia a prácticas los días de exposición es obligatoria.
- Se debe entregar de cada trabajo una memoria detallada.



Algoritmo: Definición

- ♦ *(matemáticas, ciencias de la computación y disciplinas relacionadas, del griego y latín, dixit algorithmus y este a su vez del matemático persa Al-Juarismi)*
- ♦ Conjunto de instrucciones bien definidas, ordenadas y finitas que permite realizar una actividad mediante pasos sucesivos que no generen dudas a quien deba realizar dicha actividad.
- ♦ Dados un estado inicial y una entrada, siguiendo los pasos sucesivos se llega a un estado final y se obtiene una solución.
- ♦ Ejemplo: Dado un conjunto de enteros, encontrar si existe al menos un par de ellos cuya suma sea k .

Descripción de un algoritmo

- ♦ Debemos ser muy cuidadosos con el nivel de detalle:
 - ♦ NO hay que describirlo todo con precisión, es tarea del programador.
 - ♦ NO hay que tratar el problema como una caja negra.
- ♦ Las principales partes en la descripción de un algoritmo son:
 - ♦ **Propósito** (cuál es el resultado de ejecutar el algoritmo),
 - ♦ **Estructuras de datos** (qué es manipulado por el algoritmo),
 - ♦ **Técnica** (que pasos son los que sigue el algoritmo, cómo se resuelve el problema),
 - ♦ **Justificación** (prueba de corrección del algoritmo),
 - ♦ **Análisis** (velocidad, espacio, costo, ...).

Algoritmia

- ♦ La **Algoritmia** es la ciencia que estudia los Algoritmos, y es uno de los pilares de la informática (Ciencia de la Computación en inglés).
- ♦ El estudio de los algoritmos nos permite centrarnos en el problema así como en las distintas técnicas para su resolución, independientemente de aspectos concretos de implementación (lenguaje, paradigma, hardware,...)
- ♦ Es importante en informática ya que la calidad de un software depende de:
 - 1 Los algoritmos seleccionados
 - 2 La eficiencia y adecuación de la implementación

Objetivo General

Diseñar o seleccionar el algoritmo adecuado para resolver un problema e implementarlo

- ♦ Necesitamos:
 - ♦ Conocer las distintas técnicas de diseño de algoritmos.
 - ♦ Conocer cómo éstas técnicas se aplican sobre un conjunto de problemas bien definidos, sus puntos fuertes y sus debilidades.
 - ♦ Conocer mecanismos para medir la eficiencia de los algoritmos.

Competencias específicas

Rama 6:

- Conocimiento y aplicación de los procedimientos algorítmicos básicos de las tecnologías informáticas para diseñar soluciones a problemas, analizando la idoneidad y complejidad de los algoritmos propuestos.

Pero....

Problemas en apariencia similares requieren aproximaciones muy distintas !!!

Ejemplo: Parque de Atracciones

- ◆ Disponemos de un conjunto de atracciones

A_1, \dots, A_n

- ◆ Cada atracción tiene un tiempo de inicio y otro de fin:

$\text{in}(A_i), \text{fin}(A_i)$



Ejemplo: Parque de Atracciones

- ◆ Problema I:
 - ◆ Seleccionar el mayor número de atracciones que un individuo pueda visitar.
- ◆ Problema II:
 - ◆ Seleccionar las atracciones que permitan que un visitante esté ocioso el menor tiempo posible.
- ◆ Problema III:
 - ◆ Conocidas las valoraciones de los usuarios, $val(A_i)$, seleccionar aquéllas que nos garanticen maximizar la valoración conjunta en nuestra estancia.

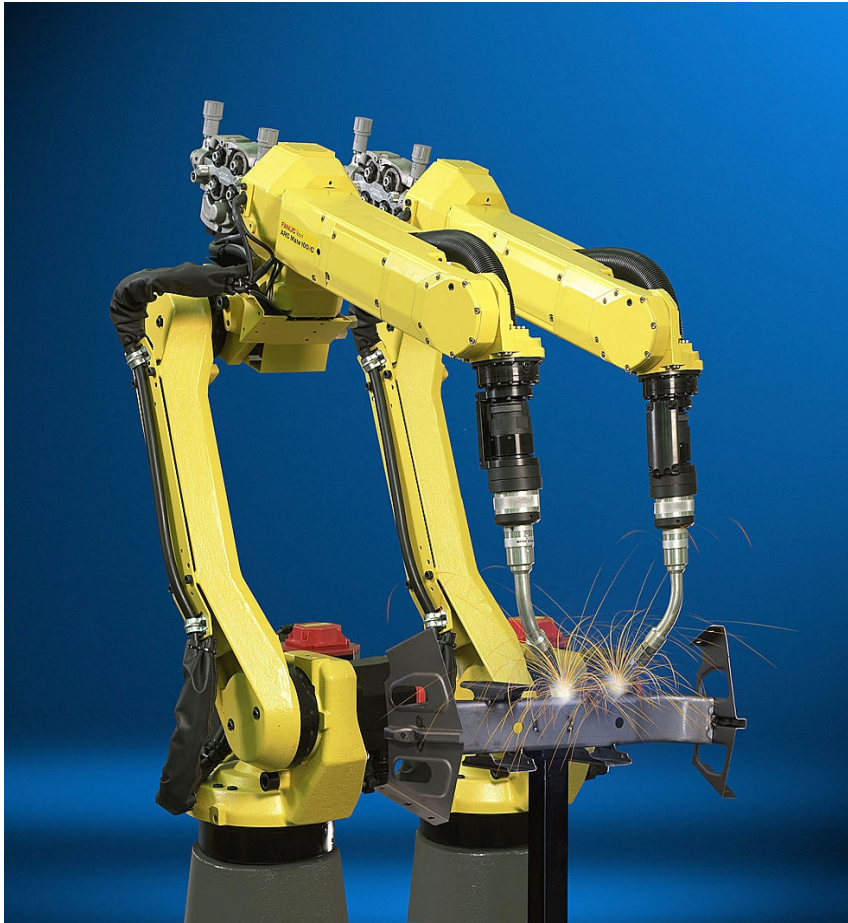
Ejemplo

- ♦ Se tienen n bolas de igual tamaño, todas ellas de igual peso salvo una más pesada. Como único medio para identificar esta bola singular se dispone de una balanza romana clásica (sólo indica si algo pesa más, menos o igual que otra cosa).



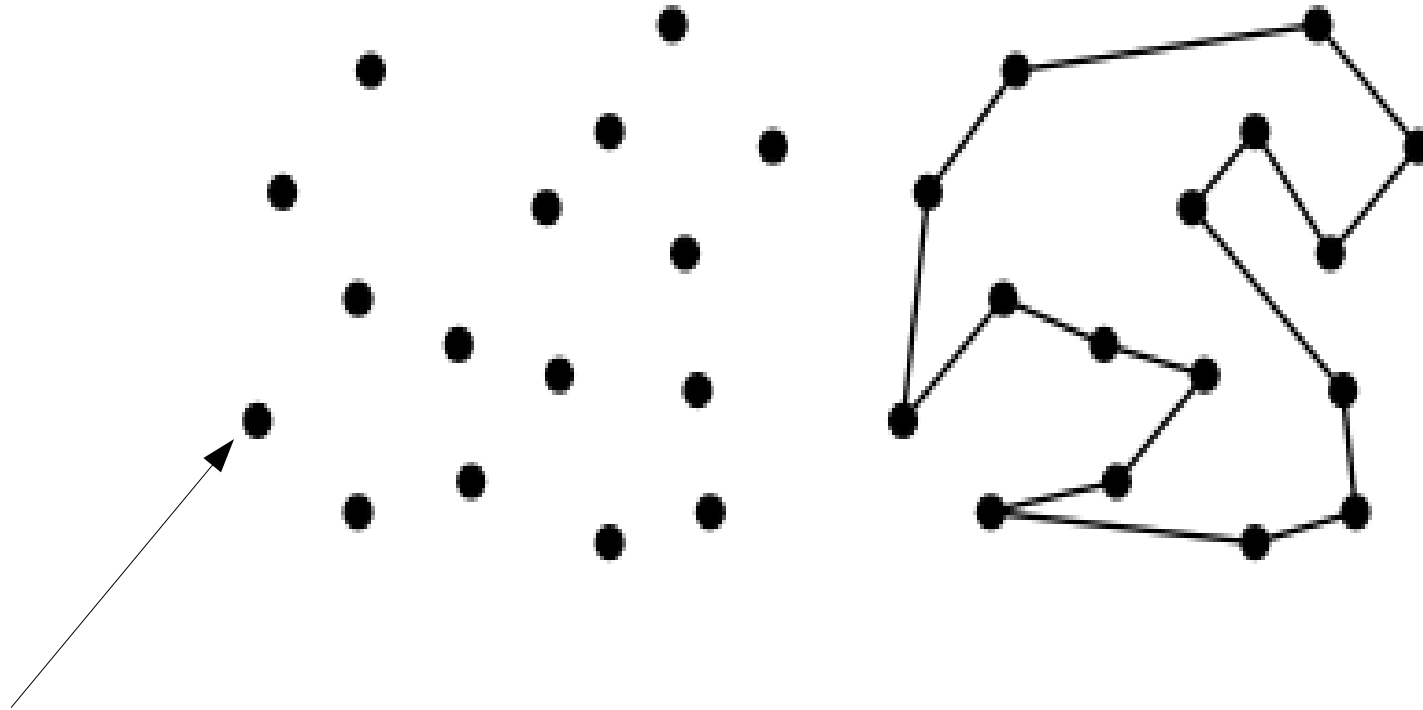
Diseñar un algoritmo que permita determinar cuál es la bola más pesada, con el mínimo número posible de pesadas.

Ejemplo : Optimizar robot



- ◆ El robot debe soldar **n puntos** en un plano.
- ◆ Buscamos el **orden** en el cual el robot debe soldar los puntos
- ◆ Criterio: **Minimizar el tiempo** en realizar la operación (depende de la distancia)

Ejemplo: Vecino más cercano



Empezar en un punto p_0 y avanzar a su vecino más cercano p_1 , entonces buscar el vecino más cercano de p_1 , etc. hasta terminar

Solución: Vecinos más cercanos

Seleccionar un punto de partida p_0

$p = p_0$

$i=0$

While existan puntos no visitados

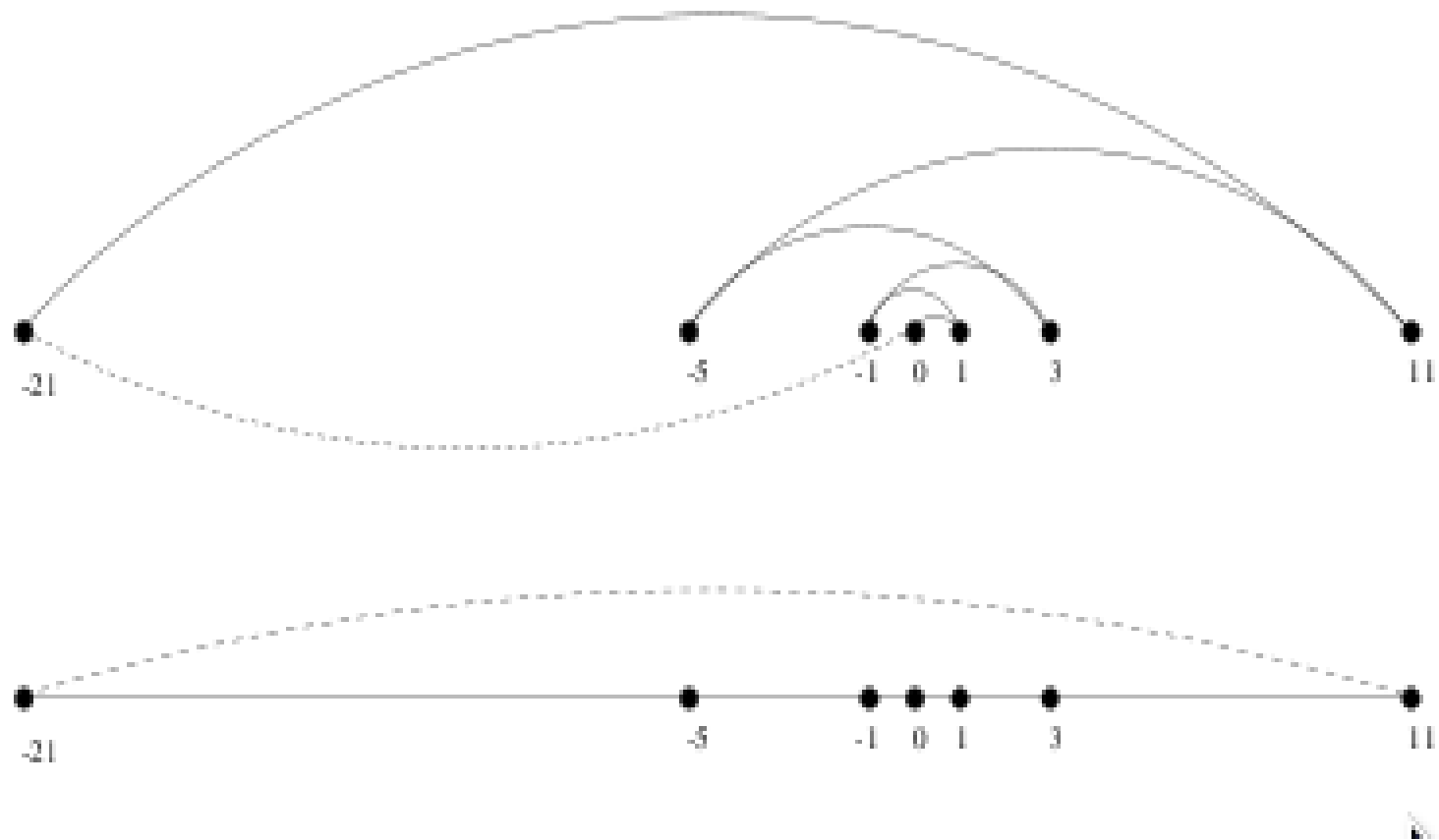
$i=i+1$

Sea p_i el punto más cercano no visitado de
 p_{i-1}

Visitar p_i

Return p_0, p_1, \dots, p_n

El criterio es erróneo !!!



Solución:

- ♦ Buscar todos los posibles ciclos y encontrar el de menor costo!!!!
- ♦ Equivale a encontrar todos los posibles ordenes de puntos

$d = \infty$

Para cada una de las $n!$ Permutaciones, P_i , de n puntos

If $(\text{cost}(P_i) \leq d)$ then

$d = \text{cost}(P_i)$ and $\text{Solucion} = P_i$

Return Solucion



Ineficiente !!!!

Objetivos (1)

- Plantearse la búsqueda de varias soluciones distintas para un mismo problema y evaluar la bondad de cada una de ellas.
- Tomar conciencia de la importancia del análisis de la eficiencia de un algoritmo como paso previo a su implementación en un lenguaje de programación.
- Conocer la notación asintótica para describir la eficiencia de un algoritmo, distinguiendo entre los distintos tipos de análisis que se pueden realizar: caso más favorable, más desfavorable y promedio.
- Saber realizar el análisis de eficiencia de un algoritmo, tanto a nivel teórico como empírico, y saber contrastar resultados experimentales con los teóricos.
- Conocer las técnicas básicas de resolución de ecuaciones de recurrencia: expansión de la recurrencia, método de la ecuación característica y utilización de fórmulas maestras.
- Comprender la técnica de resolución de un problema por división en problemas más pequeños.
- Conocer y saber aplicar los esquemas básicos de los algoritmos divide y vencerás.

Objetivos (2)

- Comprender la técnica de Avance Rápido y los distintos casos que se pueden presentar en la resolución de problemas por esta técnica: obtención de la solución óptima, de una no óptima, o no obtención de la solución.
- Comprender la técnica de la Programación Dinámica, e identificar las diferencias con Divide y Vencerás y con Avance Rápido.
- Saber identificar problemas que cumplen el Principio de Optimalidad y qué es necesario para poder aplicar esta técnica.
- Saber ver al árbol de estados como una representación lógica del conjunto de todas las posibles soluciones de un problema.
- Conocer las técnicas de exploración de grafos (vuelta atrás y ramificación y poda) y su aplicación en la resolución de problemas, entendiendo sus características principales y las diferencias entre ellas.
- Comprender y saber aplicar el uso de cotas para reducir el espacio de búsqueda en las técnicas de exploración en grafos.
- Conocer los criterios de aplicación de cada una de las distintas técnicas de diseño de algoritmos.
- Saber seleccionar e implementar el mejor algoritmo que resuelve un problema dado.

Temario

Teoría

1. La Eficiencia de los Algoritmos.
2. Algoritmos Divide y Vencerás.
3. Algoritmos Voraces.
4. Algoritmos para la Exploración en Grafos.
5. Algoritmos Basados en Programación Dinámica.
6. Otras Metodologías Algorítmicas.

Prácticas

Resolución de problemas propios de la materia

- Notación y eficiencia de Algoritmos
- Algoritmos basados en las Técnicas Divide y Vencerás, Greedy, Programación Dinámica y de Exploración en Grafos
- Resolución de problemas utilizando la técnica más adecuada.

Bibliografía

- ♦ **G. BRASSARD, P. BRATLEY.** Fundamentos de Algoritmos. Prentice Hall (1997)
- ♦ **T.H. CORMEN, C.E. LEISERSON, R.L. RIVEST, C. STEIN.** Introduction to Algorithms. The MIT Press (2001, 2009)
- ♦ **E. HOROWITZ, S. SAHNI, S. RAJASEKERAN.** Computer Algorithms. Silicon Press (2008)
- ♦ **J.L. VERDEGAY.** Curso de Teoría de Algoritmos (2004)
- ♦ **J. KLEINBERG, E. TARDOS.** Algorithm Design. Pearson (2009, 2014)
- ♦ **S. SKIENA.** The Algorithm Design Manual. Springer (2008)

Método de Evaluación

Junio:

- Un examen al final del cuatrimestre. (T)
- Prácticas a realizar a lo largo del curso. (P)
- Evaluación continua (C)
- ♦ La calificación final será $C = 0,6T + 0,4(P+C)$

Método de Evaluación

Septiembre:

- Un examen escrito que se puntuará de 0 a 10. (E)
- La calificación será $C = \text{máximo}\{C1, C2\}$:
- $C1 = E$
- $C2 = 0,6E + 0,4(P+C)$ siendo P y C la calificación obtenida durante el curso.