
Algorithms for Problem Solving – 11650

Jon Ander Gómez Adrián
jon@dsic.upv.es

Departament de Sistemes Informàtics i Computació
Escola Tècnica Superior d'Enginyeria Informàtica
Universitat Politècnica de València

2 de febrero de 2015

Presentación

- ▷ Presentación
- Objetivos I
- Objetivos II
- Contenidos
- Temario
- Bibliografía
- Evaluación
- Partida
- Webs
- Envíos
- Problemas
- Australian Voting

Asignatura: 11650
Acrónimo: CACM
Nombre: Competición de Programación
Algorithms for Problem Solving

Carga lectiva: 4.5 ECTS (3 + 1.5)

Profesor: Jon Ander Gómez Adrián
`jon@dsic.upv.es`
`http://www.dsic.upv.es/~jon`

Despacho: 314 tercer piso 1H – ETSInf
3D11 tercer piso 1F – DSIC

Tutorías: *<ver en la Web anterior>*

Objetivos I

- Presentación
- ▷ Objetivos I
- Objetivos II
- Contenidos
- Temario
- Bibliografía
- Evaluación
- Partida
- Webs
- Envíos
- Problemas
- Australian Voting

Esta asignatura se plantea como iniciación a la resolución eficiente de problemas mediante algoritmos.

Muchos de los problemas reales requieren la combinación de diferentes técnicas algorítmicas y estructuras de datos avanzadas.

Para la mayoría de estos problemas la única solución válida en un escenario real es la más eficiente posible.

Desde el punto de vista formativo:

- Cubrir la necesidad del alumnado de informática en cuanto a dominar técnicas algorítmicas avanzadas.
- Aumentar las posibilidades de salida laboral de los titulados.

Objetivos II

- Presentación
- Objetivos I
- ▷ Objetivos II
- Contenidos
- Temario
- Bibliografía
- Evaluación
- Partida
- Webs
- Envíos
- Problemas
- Australian Voting

Objetivos generales al final del curso:

- El alumno será capaz de diseñar algoritmos para la resolución de problemas complejos.
- El alumno combinará hábilmente una o más estrategias algorítmicas para aplicarlas en las soluciones a problemas.
- El alumno tendrá la capacidad de usar múltiples tipos de datos y de relacionarlos entre sí en la implementación de los algoritmos.

- Algoritmos avanzados para trabajar con árboles y grafos
- Programación dinámica
- Geometría computacional
- Teoría de números, Aritmética entera
- Manipulación de números grandes (precisión arbitraria)
- Profundización en estrategias algorítmicas: *backtracking*, *branch and bound*, A^* , IDA^* , etc.

Temario

Presentación
Objetivos I
Objetivos II
Contenidos
▷ Temario
Bibliografía
Evaluación
Partida
Webs
Envíos
Problemas
Australian Voting

1. Introducción a las competiciones de programación
2. Ordenación (repaso a sus diferentes variantes)
3. Aritmética y Álgebra
4. Teoría de números
5. Rastreo exhaustivo (*backtracking*)
6. Recorridos en grafos
7. Algoritmos sobre grafos
8. Programación dinámica
9. Mallados (*grids*)
10. Geometría y Geometría Computacional

- **Programming Challenges: The Programming Contest Training Manual**
Skiena, S.S. and Revilla, M.A.
Springer 2003
- **The Algorithm Design Manual (2ª ed.)**
Steven Skiena
Springer 2008
- **Introduction to Algorithms (3ª ed.)**
Thomas H. Cormen, Clifford Stein, Ronald L. Rivest and Charles E. Leiserson
McGraw-Hill Higher Education 2001
- **Fundamentos de Algoritmia**
G. Brassard and P. Bratley
Prentice-Hall 1997

- Evaluación continua. No hay exámenes.
- 10 problemas para poder aprobar, 1 por semana.
- <http://www.programming-challenges.com>
- Nivel de los problemas 2 o superior.
- El profesor llevará el seguimiento de los problemas que vaya resolviendo cada alumno.
- Se puede y se recomienda trabajar en grupo.
- Nota final en función del número de problemas resueltos y la dificultad de estos.
- M.H. sólo puede haber una.

De entrada se asume que todos los alumnos conocen

- listas, pilas y colas
- colas con prioridad: heaps
- diccionarios: tablas HASH y Trie (árbol aceptor de prefijos)
- conjuntos: MFSET u otras implementaciones
- árboles y grafos

Webs recomendadas

- Presentación
- Objetivos I
- Objetivos II
- Contenidos
- Temario
- Bibliografía
- Evaluación
- Partida
- ▷ Webs
- Envíos
- Problemas
- Australian Voting

- <http://www.dsic.upv.es/users/clocalprog>
- <http://www.programming-challenges.com>
- <http://uva.onlinejudge.org>
- <http://icpc.baylor.edu>
- <http://train.usaco.org>

Envíos al juez automático

- Presentación
- Objetivos I
- Objetivos II
- Contenidos
- Temario
- Bibliografía
- Evaluación
- Partida
- Webs
- ▷ Envíos
- Problemas
- Australian Voting

- Crear usuario en el juez de la UVa y en el Programming Challenges
- Envío de un problema ejemplo para ver la mecánica de los jueces automáticos

Problemas para comenzar

- Presentación
- Objetivos I
- Objetivos II
- Contenidos
- Temario
- Bibliografía
- Evaluación
- Partida
- Webs
- Envíos
- ▷ Problemas
- Australian Voting

- Todos los del tema 1 del libro "Programming Challenges"
- Los problemas 1, 2, 3, 5 y 7 del tema 2

Problemas para comenzar

Presentación
Objetivos I
Objetivos II
Contenidos
Temario
Bibliografía
Evaluación
Partida
Webs
Envíos
▷ Problemas
Australian Voting

- Todos los del tema 1 del libro "Programming Challenges"
- Los problemas 1, 2, 3, 5 y 7 del tema 2
- Trabajad el problema **110206/10044 "Erdős numbers"** para la próxima clase
- Este problema debe utilizar
 - un diccionario, tabla HASH o Trie, para ser rápido, y
 - un algoritmo estilo BFS de recorrido de grafos para obtener la solución
- Tened especial atención a la lectura de datos

Australian Voting

- Presentación
- Objetivos I
- Objetivos II
- Contenidos
- Temario
- Bibliografía
- Evaluación
- Partida
- Webs
- Envíos
- Problemas
- ▷ Australian Voting

1. Inicialmente todos los candidatos están activos
2. Poner a cero el contador de votos de cada candidato
3. Contar los votos de todos los activos según orden de preferencia en cada papeleta
4. Si el candidato con más votos tiene más del 50 % se muestra su nombre y **finalizar**
5. Si el candidato con más votos tiene el mismo número de votos que el candidato con menos votos, mostrar todos los activos y **finalizar**
6. Eliminar aquellos candidatos cuyo número de votos sea igual al mínimo
7. Volver al punto 2
8. Finalizar