

Nom de l'assignatura : Metodologia i tecnologia de la programació II

Codi	Tipus	Curs/semestre	Crèdits ECTS
21307	<ul style="list-style-type: none"> • Troncal • Semestral 	2n curs / 1er semestre	6

Professors

Nom	Dpt/Unitat	Despatx	Direcció e-mail	Telèfon
Ramon Baldrich	Ciències de la Computació	S/246	Ramon.baldrich@uab.cat	935811828
Alicia Fornés	Ciències de la Computació	S/247	alicia.fornes@uab.cat	935811828
Josep M. Gonfaus	Ciències de la Computació	S/247	gonfaus@cvc.uab.cat	935811828

Objectius

Objectiu genèric de l'assignatura

Proveir a l'alumne del coneixement necessari per tal d'abordar problemes de complexitat major, mitjançant tècniques que li permetin analitzar un algorisme (correcció, cost, ...), i dissenyar noves solucions a partir de la idea de recursivitat i/o de tècniques de disseny d'algorismes.

Coneixements

1. PARADIGMA DE LA PROGRAMACIÓ ORIENTADA A OBJECTE
2. RECURSIVITAT
 - a. Metodologia de disseny d'algorismes recursius. Prova de la correcció i l'acabament. Exemples: factorial i exponencial de nombres naturals.
 - b. Anàlisi d'algorismes recursius: càlcul de la complexitat. Equacions de recurrència. Exemples: Merge-sort, multiplicació de naturals.
 - c. Implementació dels algorismes recursius fent servir piles de recursió. Concepte de registre d'activació.
 - d. Tècniques de transformació recursiva-iterativa d'algorismes. Eliminació de la recursivitat final i no final. Recursivitat múltiple. Exemples.
3. COMPLEXITAT DELS ALGORISMES
 - a. Introducció. Criteris per seleccionar un algorisme. Factors que influeixen en el temps d'execució.
 - b. Exemple: càlcul del cost d'un algorisme d'ordenació per inserció.
 - c. Notació asimptòtica: Theta-notació, O-notació, omega-notació.
 - d. Càlcul del temps d'execució d'un programa. Suma i multiplicació en notació asimptòtica. Sugerències pràctiques pel càlcul de la complexitat.
 - e. Comparació entre les complexitats més usals.
4. TÈCNiques DE DISSENY D'ALGORISMES
 - a. Introducció.
 - b. Tècnica incremental. Exemple: Ordenació per inserció.
 - c. Mètodes de Backtracking i Branch & Bound .
 - d. Backtracking. Exemple: problema de les n reines.
 - e. Branch & Bound. Exemple: puzzle de 15 peces.
 - f. Divide and Conquer. Esquema general de la tècnica. Exemples: Merge-sort, les torres de Hanoi. Equacions de recurrència associades.
 - g. Programació dinàmica. Idea general del mètode. Exemple: trobar el camí més curt entre ciutats.
 - h. Tècniques Greedy. Esquema general. Exemple: problema de la motxilla.
 - i. Programació no determinista. Mètodes de Montecarlo i Las Vegas

Habilitats

- Entendre i interpretar enunciats de problemes computacionals complexos.
- Treballar amb un llenguatge algorísmic independent de programació.
- Disenyar algorismes recursius i saber-ne fer el seguiment amb crides múltiples.
- Descripció de la solució al problema i la estructura que el suporta.
- Avaluar els costos del problema respecte a diferents recursos.
- Dissenyar algorismes a partir de paradigmes avançats.
- Destriar el millor paradigma donat la estructura del problema.

Competències

1. Raonament analític.
2. Resolució de problemes.
3. Abstracció.
4. Capacitat d'organització i planificació.

Capacitats prèvies

Algorísmica bàsica
Especificació de problemes simples (ordenació, cerca, ...).
Desenvolupament algebraic.
Nivell avançat de programació.

Continguts

(T:teoria, S:seminaris=reunions tutoritzades i fites, PS:preparació de seminaris, L:laboratoris, PP:preparació pràctiques, E:estudi, AA:altres activitats)

1. Presentació de la assignatura	T	S	PS	L	E	PP	AA	Total
	1							1

Explicació del funcionament de la assignatura: planificació, avaluació, temari, pràctiques a desenvolupar.

2. Paradigma de la programació OO	T	S	PS	L	E	PP	AA	Total
	4			2				6

Introducció a paradigma Orientat a Objecte utilitzant el llenguatge JAVA com a exemple d'ús.

3. Recursivitat	T	S	PS	L	E	PP	AA	Total
	4	2		2				8

Metodologia de disseny d'algorismes recursius. Prova de la correcció i l'acabament. Anàlisi d'algorismes recursius: càlcul de la complexitat. Equacions de recurrència. Implementació dels algorismes recursius fent servir piles de recursió. Concepte de registre d'activació. Tècniques de transformació recursiva-iterativa d'algorismes. Eliminació de la recursivitat final i no final. Recursivitat múltiple.

4. Complexitat	T	S	PS	L	E	PP	AA	Total
	4	2		4				10

Criteris per seleccionar un algorisme. Factors que influeixen en el temps d'execució. Notació asimptòtica: Theta-notació, O-notació, omega-notació. Càlcul del temps d'execució d'un programa. Suma i multiplicació en notació asimptòtica. Sugeriments pràctics pel càlcul de la complexitat. Comparació entre les complexitats més usuals.

5. Tècniques de disseny d'algorismes	T	S	PS	L	E	PP	AA	Total
	16	6		12				34,0

Mètodes de Backtracking i Branch& Bound . Backtracking. Divide and Conquer. Esquema general de la tècnica. Equacions de recurrència associades. Programació dinàmica. Idea general del mètode. Tècniques Greedy. Esquema general. Programació no determinista: Mètodes de Montecarlo i Las Vegas.

Metodologia docent

La metodologia docent de l'assignatura es basa en impartir classes magistrals utilitzant transparències. L'alumne pot disposar d'aquest material en aquesta pàgina prèviament a la classe.

Per la part de pràctiques les classes s'imparteixen en sessions de laboratori, tutelades per professor.

Avaluació

Avaluació continuada	Examen final (1ª convocatòria)	Examen final (2ª convocatòria)
Teoria: NO Pràctiques: SI	Teoria: Obligatori per a tothom Pràctiques: Obligatori pels qui no hagin superat la avaluació continuada	Només per a als estudiants que compleixin les condicions següents: No hagin superat la 1ª convocatòria en alguna de les seves parts

Nota Final = nota examen teoria * 0.6 + nota pràctiques * 0.4

Nota pràctiques = nota laboratori * 0.5 + nota projectes * 0.5

Cada una de les notes estan en el rang 0-10.

Nota laboratori: s'avaluarà de forma continuada la participació en les sessions de pràctiques, tenint, principalment, en compte la pro-activitat.

Hi haurà 10 sessions avaluables.

Nota projectes: s'avaluarà la consecució dels objectius plantejats en cada un dels projectes.

Planificació temporal

Horaris de classe

Teoria:

Dilluns de 09:00 a 11:00

Dilluns de 15:00 a 17:00

Problemes:

Divendres de 11:00 a 12:00

Divendres de 17:00 a 18:00

Pràctiques:

- Dilluns de 11:00 a 13:00
- Dilluns de 13:00 a 15:00
- Dijous de 15:00 a 17:00
- Dijous de 17:00 a 19:00
- Dijous de 19:00 a 21:00
- Divendres de 9:00 a 11:00

Setmanes de problemes

- Octubre 15, 29
- Novembre 5, 12, 19, 26
- Desembre 3, 10, 17
- Gener 14

Setmanes de pràctiques

Grups de dilluns	Grups de Dijous	Grups de Divendres
Octubre 18, 25	Octubre 21, 28	Octubre 22, 29
Novembre 8, 15, 22, 29	Novembre 11, 18, 25	Novembre 12, 19, 26
Desembre 13,20	Desembre 2, 9, 16	Desembre 3, 10, 17
Gener 10, 17	Gener 13, 20	Gener 14, 21

Bibliografia bàsica

- Aho, J. Hopcroft, J. Ullman. Estructurasde datos y algoritmos, Addison Wesley, 1987.
- G. Brassard, P. Bratley. Fundamentosde algoritmia, Prentice Hall, 1997.
- T. Cormen, C. Leiserson, R. Rivest. Introductionto algorithms, Mc Graw Hill , 1990.
- E. Horowitz, S. Sahni. Fundamentals of computer algorithms, Computer Science Press, 1989.

Bibliografia complementària

- J. Arsac. Las bases de la programación, Ed. Omega, 1986.
- S. Baase. Computer Algorithms. Introduction to Design and Analysis, Addison-Wesley Publishing Company, 2nd edition.
- X. Franch Gutiérrez. Estructuras de dades. Especificació, disseny i implementació, Edicions UPC, 1993.
- L. Joyanes Aguilar. Fundamentosde programación (Algoritmos y Estructuras de datos), Mc Graw Hill; edició, 1988.
- L. Kronsjö. Algorithms: Their Complexity and Their Efficiency, Ed. John Wiley & Sons.
- N. Wirth. Algoritmosy Estructuras de Datos, Prentice Hall, 1986.

Enllaços web

Inclosos en cada tema al Caront.