

Estructures de Dades

Codi: 104347
Crèdits: 6

Titulació	Tipus	Curs	Semestre
2503758 Enginyeria de Dades	OB	2	1

La metodologia docent i l'avaluació proposades a la guia poden experimentar alguna modificació en funció de les restriccions a la presencialitat que imposin les autoritats sanitàries.

Professor/a de contacte

Nom: Gemma Sánchez Albaladejo
Correu electrònic: Gemma.Sanchez@uab.cat

Utilització d'idiomes a l'assignatura

Llengua vehicular majoritària: català (cat)
Grup íntegre en anglès: No
Grup íntegre en català: Sí
Grup íntegre en espanyol: No

Equip docent

Gemma Sánchez Albaladejo
Daniel Soto Alvarez

Prerequisits

L'assignatura no té cap prerequisit oficial. De totes formes, s'assumeix que l'estudiant ha cursat les assignatures prèvies de Fonaments de programació i Programació Avançada, així com Grafs, Topologia i Geometria Discreta. Per tant, està familiaritzat amb les estructures bàsiques i avançades de la programació, Orientació a objectes i el concepte de graf amb els diferents mètodes de recorregut sobre ells.

Objectius

Aquesta assignatura forma part de la matèria Representació de les Dades i s'ha de veure com la continuació lògica de l'assignatura Programació Avançada i la continuació pràctica de l'assignatura de Matemàtica Discreta. L'objectiu bàsic és aprofundir en les estructures de dades bàsiques introduïdes a Fonaments de programació junt amb les nocions de programació orientada a objectes introduïdes a Programació Avançada i ampliar-les amb altres estructures de dades més complexes així com algorismes eficients per recorre-les. S'introduirà el concepte d'algorisme recursiu amb algorismes recursius simples i més complexos com els relacionats amb recorreguts d'arbres i grafs

D'aquesta forma, els objectius formatius que es proposen per a l'assignatura són els següents:

- Ser capaç d'analitzar un problema complex, dissenyar una solució òptima, implementar-la, calcular el seu cost i provar-la.
- Entendre i saber utilitzar estructures de dades complexes com arbres, grafs etc. i utilitzar-les correctament i d'una manera eficient per resoldre problemes algorísmics complexos.
- Dotar l'alumne de la capacitat de disseny d'algorismes per a la resolució de problemes complexos, veient algorismes complexos de recorregut i cerca en estructures de dades complexes. A més d'analitzar la complexitat temporal i espacial d'ells per tal de triar la solució que més s'adapti a les necessitats de cada moment.

- Introduir el concepte de recursivitat i la seva aplicació al recorregut d'estructures complexes recursives, a més de ser capaç d'analitzar la complexitat d'algorismes recursius.
- Programar en un llenguatge de programació real i ser capaç de depurar els propis programes.
- Desenvolupar els programes seguint unes normes d'estil tendents a aconseguir programes de qualitat.

Competències

- Buscar, seleccionar i gestionar de manera responsable la informació i el coneixement.
- Desenvolupar un pensament i un raonament crític i saber comunicar-los de manera efectiva, tant en les llengües pròpies com en anglès.
- Dissenyar solucions algorítmiques eficients per a problemes computacionals, implementar-les en forma de desenvolupaments de programari robust, estructurat i fàcil de mantenir, i verificar-ne la validesa.
- Treballar cooperativament, en entorns complexos o incerts i amb recursos limitats, en un context multidisciplinari, assumint i respectant el rol dels diferents membres de l'equip.

Resultats d'aprenentatge

1. Buscar, seleccionar i gestionar de manera responsable la informació i el coneixement.
2. Desenvolupar programes amb un bon estil de programació i ben documentats i saber depurar-los, testar-los i corregir-los.
3. Desenvolupar un pensament i un raonament crític i saber comunicar-los de manera efectiva, tant en les llengües pròpies com en anglès.
4. Seleccionar i aplicar la combinació d'estructures de dades i estratègies de resolució més apropiada per resoldre de manera eficient un problema informàtic.
5. Treballar cooperativament, en entorns complexos o incerts i amb recursos limitats, en un context multidisciplinari, assumint i respectant el rol dels diferents membres de l'equip.

Continguts

0. Introducció

Objectius i presentació de l'assignatura. Repàs de Programació avançada.

1. Estructures de dades lineals. Llistes, cues, piles.

Representació i manipulació d'estructures de dades dinàmiques: repàs de llistes, introducció a les piles i cues.

2. Estructures de dades no lineals. Hash

Tècniques de "Hashing". Llistes hash. Funcions hash. Com s'implementen en python. Diccionaris.

3. Recursivitat i Algorismes d'ordenació

Introducció als algorismes recursius. Mètode de la bombolla, QuickSort, MergeSort. Recursivitat. Càlcul complexitat.

4. Estructures de dades no lineals. Grafs

Representacions i recorreguts. BFS, DFS, Resolució de problemes amb grafs.

5. Estructures de dades no lineals. Arbres.

Definició i representació d'un arbre. Recorreguts d'arbres. Binary Heaps.

Metodologia

NOTA INFORMATIVA DEL CURS DEGUT AL COVID: Aquesta assignatura es farà en mode on-line degut a la pandèmia, per tant totes les activitats que figuren com a presencials es faran virtuals mitjançant caronte i teams a no ser que les indicacions del govern i/o la UAB canviïn al llarg del curs.

La metodologia docent de l'assignatura parteix del principi que diu que "*programar és l'única forma d'aprendre a programar*" i, per tant, estarà centrada principalment en el treball pràctic de l'estudiant. Les sessions presencials de classe s'organitzaran per introduir els continguts teòrics de l'assignatura, des d'una perspectiva molt pràctica a partir d'exemples i d'exercicis i problemes de programació que s'hauran de resoldre a classe directament amb l'ordinador. L'objectiu principal de l'assignatura és que l'estudiant sàpiga resoldre un problema donat, de manera eficient, utilitzant estructures de dades complexes, si cal. Per aquesta raó l'aprenentatge es centrarà en acompanyar a l'estudiant en la seva tasca de resolució de problemes a partir d'uns conceptes teòrics estudiats prèviament de manera autònoma. Es farà servir principalment el llenguatge de programació python.

Per una altra banda, es realitzarà un projecte de programació que s'haurà d'anar desenvolupant de forma principalment autònoma durant tot els curs (amb seguiment i control per part del professor en sessions puntuals) i que suposarà integrar de forma pràctica gairebé tots els conceptes i eines de programació introduïts a les sessions presencials en la resolució d'un problema real complex. A més a més, es proposarà un conjunt d'exercicis que s'hauran de resoldre de forma individual al llarg del curs (alguns dels quals es resoldran i discutiran a les sessions presencials) que han de servir per comprendre, integrar i aplicar els conceptes desenvolupats a les sessions presencials. A les activitats del curs (sessions presencials, problemes i pràctiques) es farà servir principalment el llenguatge de programació Python.

A nivell presencial, les sessions de classe s'organitzaran en quatre hores setmanals i es faran en una aula amb ordinadors per facilitar el treball pràctic de l'estudiant. S'encoratja que l'alumne porti el seu propi portàtil a classe si en disposa d'un. A les sessions presencials s'aniran introduint els conceptes que es detallen al temari de l'assignatura. En alguns casos, es podran posar a disposició de l'estudiant vídeos explicatius o altre material complementari que l'estudiant haurà de visionar abans de la sessió de classe. Les sessions de classe tindran un enfoc força pràctic amb exemples i exercicis que es plantejaran als alumnes per facilitar la comprensió i aprenentatge dels conceptes explicats. Aquests exercicis es realitzaran i discutiran durant la sessió i serviran per anar introduint els continguts de l'assignatura i veure'n la seva aplicació pràctica.

L'estudiant haurà de completar les classes presencials amb el treball personal autònom en la realització dels exercicis que es vagin proposant i que han de servir per acabar d'entendre els continguts de l'assignatura. Cal tenir present que el temari de l'assignatura té una continuïtat lògica al llarg del curs, de manera que per poder seguir correctament una classe cal haver assimilat el que s'ha explicat a les sessions anteriors. Alguns d'aquests exercicis s'hauran de lliurar de forma individual com a part de l'avaluació de l'assignatura.

A més a més, els estudiants hauran de fer en grups de 2 un projecte de programació que es desenvoluparà de forma autònoma durant tot el curs fora de les sessions presencials. El projecte de programació permetrà abordar un problema de programació d'una certa complexitat que integri la majoria dels conceptes explicats durant el curs. Durant el curs, es dedicaran algunes sessions presencials al control, seguiment i avaluació del treball fet per l'alumne en el projecte de programació.

La gestió de la docència de l'assignatura es farà a través de Caronte (<http://caronte.uab.cat/>), que servirà per poder veure els materials, gestionar els grups de pràctiques, fer els lliuraments corresponents, veure les notes, comunicar-se amb els professors, etc.

La gestió de les classes presencials que degut a la pandèmia es facin on-line es farà a través de teams <https://teams.microsoft.com/>, qualsevol canvi al llarg del curs es notificarà a Caronte.

Nota: es reservaran 15 minuts d'una classe, dins del calendari establert pel centre/titulació, per a la complementació per part de l'alumnat de les enquestes d'avaluació de l'actuació del professorat i d'avaluació de l'assignatura/mòdul.

Activitats formatives

Títol	Hores	ECTS	Resultats d'aprenentatge
Tipus: Dirigides			
Classes presencials	48	1,92	1, 2, 3, 4, 5

Avaluació

L'avaluació de l'assignatura tindrà en compte tres tipus d'activitats d'avaluació: lliurament de problemes, avaluació individual i projecte de programació. La nota final de l'assignatura s'obté combinant l'avaluació d'aquestes 3 activitats de la manera següent:

$$\text{Nota Final} = (0.2 * \text{Avaluació Problemes}) + (0.4 * \text{Projecte}) + (0.4 * \text{Avaluació Individual})$$

- Lliurament de problemes: en aquest apartat s'inclou el lliurament dels exercicis que es proposin al llarg del curs i altres activitats que es realitzin a les sessions de problemes.

No es necessari aconseguir una nota mínima en aquesta activitat per poder aprovar l'assignatura.

Els exercicis que es lliurin fora de termini o que tinguin una avaluació de suspès es podran recuperar i tornar a lliurar en qualsevol moment del curs abans de la data de l'examen final de l'assignatura, amb una reducció sobre la nota del 20%. Els problemes estaran ponderats segons el pes del tema al conjunt de l'assignatura, i el nombre de problemes que s'hagin de lliurar per cada tema.

- Avaluació individual: en aquest apartat s'inclou el resultat de les proves individuals que es faran al llarg del curs. Hi haurà dues proves parcials que es faran durant el període lectiu del curs en horaris de classe i una prova final durant el període oficial d'exàmens. Aquesta prova final serà de recuperació i només l'haurà de fer els estudiants que no hagin superat algun dels dos parcials. Si s'ha superat un dels dos parcials, però l'altre no, en aquesta prova només s'ha de recuperar la part de l'assignatura corresponent al parcial que no s'hagi superat.

S'haurà d'aconseguir una nota mínima de 4 en cadascun dels dos parcials i una nota promig mínima de 5 per poder aprovar l'assignatura .

$$\text{La nota final serà la mitja dels dos parcials: Avaluació Individual} = (0.5 * \text{Parcial1}) + (0.5 * \text{Parcial2})$$

- Projecte: inclou tot el treball del projecte de programació. Inclou l'avaluació dels dos lliuraments del projecte (un lliurament parcial a meitat de curs i el lliurament final) i l'avaluació del seguiment del projecte que es farà a les sessions presencials que correspongui. La nota final es calcularà de la forma següent:

$$\text{Projecte} = (0.2 * \text{Avaluació seguiment projecte}) + (0.3 * \text{Entrega Parcial 1}) + (0.5 * \text{Entrega Final})$$

- S'haurà d'aconseguir una nota mínima de 4 en l'avaluació del seguiment del projecte i una nota mínima de 5 en el lliurament final del projecte per poder aprovar el projecte.
- S'haurà d'aconseguir una nota mínima de 5 en el projecte per poder aprovar l'assignatura.
- La nota de l'entrega final del projecte es podrà recuperar si la nota del projecte és ≥ 3 i la nota de l'avaluació individual és ≥ 5 .

No avaluable: Un alumne es considerarà no avaluable (NA) si no fa com a mínim el 50% dels lliuraments d'exercicis i no fa cap de les proves d'avaluació següents: parcial 1, parcial 2, prova final de recuperació, lliurament final de la pràctica.

Suspesos: Si el càlcul de la nota final és igual o superior a 5 però no s'arriba al mínim exigít en alguna de les activitats d'avaluació, la nota final serà suspès i es posarà un 4.5 a la nota de l'expedient de l'alumne.

Convalidacions: Pels alumnes repetidors es convalidarà la nota del projecte de l'any anterior (curs 2019-20) si es compleixen aquestes condicions:

a) La nota final del projecte del curs anterior és més gran o igual a 7

b) La nota de l'avaluació individual del curs anterior és més gran o igual a 3

MH: Es donaran tantes matrícules com es puguin dins de la normativa de la universitat, començant per les notes més altes i sempre i quan la nota mínima sigui un 9.

Revisions: Per a cada activitat d'avaluació, s'indicarà un lloc, data i hora de revisió en la que l'estudiant podrà revisar l'activitat amb el professor. En aquest context, es podran fer reclamacions sobre la nota de l'activitat, que seran avaluades pel professorat responsable de l'assignatura. Si l'estudiant no es presenta a aquesta revisió, no es revisarà posteriorment aquesta activitat.

Nota important sobre còpies i plagis:

Sense perjudici d'altres mesures disciplinàries que s'estimin oportunes, i d'acord amb la normativa acadèmica vigent, les irregularitats comeses per un estudiant que puguin conduir a una variació de la qualificació es qualificaran amb un zero (0). Les activitats d'avaluació qualificades d'aquesta forma i per aquest procediment no seran recuperables. Si és necessari superar qualsevol d'aquestes activitats d'avaluació per aprovar l'assignatura, aquesta assignatura quedarà suspesa directament, sense oportunitat de recuperar-la en el mateix curs. Aquestes irregularitats inclouen, entre d'altres:

- la còpia total o parcial d'una pràctica, informe, o qualsevol altra activitat d'avaluació;
- deixar copiar;
- presentar un treball de grup no fet íntegrament pels membres del grup;
- presentar com a propis materials elaborats per un tercer, encara que siguin traduccions o adaptacions, i en general treballs amb elements no originals i exclusius de l'estudiant;
- tenir dispositius de comunicació (com telèfons mòbils, smart watches, etc.) accessibles durant les proves d'avaluació teòric-pràctiques individuals (exàmens).

En aquests casos, la nota numèrica de l'expedient serà el valor menor entre 3.0 i la mitjana ponderada de les notes (i per tant no serà possible l'aprobat per compensació).

En l'avaluació dels lliuraments de problemes i pràctiques s'utilitzaran eines de detecció de còpia del codi del programa.

Nota sobre la planificació de les activitats d'avaluació:

Les dates d'avaluació continuada i lliurament de treballs es publicaran al principi de curs i poden estar subjectes a canvis de programació per motius d'adaptació a possibles incidències. Sempre s'informarà a Caronte sobre aquests canvis ja que s'entén que aquesta és la plataforma habitual d'intercanvi d'informació entre professors i estudiants.

Activitats d'avaluació

Títol	Pes	Hores	ECTS	Resultats d'aprenentatge
Primer Parcial	20% Nota final	2	0,08	3, 4
Problemes	20% Nota final	0	0	1, 2, 3, 4
Projecte	40% Nota final	2	0,08	1, 2, 3, 4, 5
Recuperació Final	40% Nota final	3	0,12	3, 4
Segon Parcial	20% Nota final	2	0,08	3, 4

Bibliografia

Problem solving with algorithms and data Structures using Python. Bradley N. Miller and David L. Ranum. Franklin, Beedle and associates, 2005.

Python Programming: an introduction to computer science. John Zelle. Franklin, Beedle and associates, 2004.

Data Structures and Algorithms in python. Michael T. Googrich, Roberto Tamassia, Michael H. Goldwasser. Ed. Wiley. 2013

<https://www.geeksforgeeks.org/python-programming-language/?ref=ghm>

Programari

Spyder (Anaconda)