**Algoritmica grafurilor**

Link la fișa disciplinei: https://profs.info.uaic.ro/~croitoru/ag/week01.pdf

Subiecte:

1. Grafuri bipartite: definitie, recunoastere, exemple.

2. Arbori: definitie, caracterizari.

3. Digrafuri aciclice: sortare topologica.

4. Parcurgeri sistematice in (di)grafuri.

5. Algoritmul lui Dijkstra de drum minim.

6. Problema cuplajului de cardinal maxim; complexitate timp.

7. Problema fluxului maxim. Teorema fluxului maxim - sectiunii minime.

8. Aflarea numarului de stabilitate al unui graf bipartit cu ajutorul fluxurilor.

9. Numarul de conexiune al unui graf. Algoritm polinomial de determinare.

10. Probleme NP-complete pe grafuri. Exemple.

11. Grafuri planare. Orice graf planar este un graf rar.

12. Grafuri planare. Formula lui Euler.

**Arhitectura calculatoarelor şi sisteme de operare**

Link la fișa disciplinei: http://www.info.uaic.ro/~webdata/planuri/licenta/CS1102.pdf

Subiecte:

1. Ierarhia de memorii. Memoria cache.

2. Minimizarea functiilor booleene. Diagrame Karnaugh.

3. Circuite secventiale: bistabile, latch, flip-flop.

4. Reprezentari in virgula fixa: modul-semn, complement fata de 1, complement fata de 2.

5. Operatii in virgula fixa. Depasiri.

6. Reprezentari in virgula mobila: simpla precizie, dubla, precizie.

7. Operatii in virgula mobila. Depasiri inferioare si superioare.

8. Procesoare. Conceptul pipeline.

9. Sistemul de intreruperi. Tratarea intreruperilor.

10. Arhitecturi paralele de calcul. Concepte de baza.

11. Moduri de functionare a procesorului: nucleu, utilizator. Apeluri sistem.

12. Gestiunea memoriei in sistemele de operare: adrese virtuale si fizice; segmentare, paginare.

**Baze de date**

Link la fișa disciplinei:

Link la pagina disciplinei:https://profs.info.uaic.ro/~bd/

Subiecte:

1. Relatii si operatii cu relatii in modelul relational si corespondenta in SQL (cel putin 5 operatii).

2. Dependente functionale si multivaluate: definitie si exemple.

3. Forme normale pentru scheme relationale: rolul, discutie despre BCNF sau 4NF la alegere.

4. Indexare: structuri de date utilizate si eficienta.

5. Ce este o tranzactie? Explicati proprietatile ACID.

6. Suportul SQL pentru tranzactii (inceperea si finalizarea unei tranzactii, revenirea la un punct intermediar).

7. Constrangeri asupra datelor posibil a fi declarate in BD relationale (chei si alte tipuri).

8. Tabele virtuale (views) - rol, tipuri, comportamenul la operatii de actualizare lansate asupra lor.

9. Operatia Join: join interior, exterior, left/right join, self-join.

10. Algoritmi utilizati in realizarea operatiei Join.

11. Agregarea inregistrărilor în SQL.

12. Subinterogari necorelate versus subinterogari corelate.

**Calcul numeric**

Link la fișa disciplinei:<https://profs.info.uaic.ro/~ancai/CN/fisa_CN.pdf>

Subiecte:

1. Algoritmul de eliminare Gauss pentru rezolvarea sistemelor liniare. Tehnici de pivotare
2. Descompuneri LU. Rezolvarea sistemelor triunghiulare.
3. Matrice pozitiv definite. Factorizarea Cholesky. Rezolvarea sistemelor liniare folosind factorizarea Cholesky.
4. Matrice ortogonale, matrice de reflexie. Descompuneri QR - metoda Householder, rezolvarea sistemelor liniare cu descompunerea QR
5. Metode iterative de rezolvare a sistemelor mari și rare - schema generală. Metoda Gauss-Seidel
6. Scheme economice de memorare a matricelor rare.
7. Valori si vectori proprii - definiții. Algoritmul QR.
8. Valori si vectori proprii - definiții. Metoda puterii.
9. Rezolvarea sistemelor liniare nepătratice în sensul celor mai mici pătrate. Pseudoinversa.
10. Metoda tangentei (Newton) pentru aproximarea rădăcinilor unei ecuații neliniare f(x)=0.
11. Interpolare numerică - polinomul de interpolare Lagrange.
12. Interpolare numerică - funcții spline. Funcții spline liniare continui.

**Fundamente algebrice ale informaticii**

Link la fișa disciplinei:https://profs.info.uaic.ro/~fltiplea/

Subiecte:

1.Multimi inductiv definite; functii definite recursiv; metoda inductiei structurale

2.Monoizi liber generati; coduri de lungime variabila

3. Coduri Huffman (definitii si constructie)

4. Metoda de codificare Huffman adaptiva

5.Grupuri ciclice

6. Ordinul unui element intr-un grup

7.Radacini primitive

8.Caracteristica unui inel

9.Corpuri finite: contructie

10.Spatii vectoriale

11.Coduri detectoare de erori

12.Coduri corectoare de erori

**Grafică pe calculator**

Link la fișa disciplinei: <https://profs.info.uaic.ro/~ghirvu/download/00fisagpc>

Subiecte:

1. Lumina cromatică - modelele de culori RGB, HSV.

2. Trasarea segmentelor de dreaptă pe ecrane rastru (problema, algoritmi).

3. Trasarea cercurilor pe ecrane rastru (problema, algoritmi).

4. Transformări geometrice 2D/3D (translatie, scalare) – definiție, exprimare matricială.

5. Transformări geometrice 2D/3D (rotatie) – definiție, exprimare matricială.

6. Proiecții geometrice planare (definiție, clasificare).

7. Curbe parametrice cubice. Definiție. Exemple: curbe Hermite, Bezier.

8. Algoritmi de decupare ale unor segmente de dreapta sau poligoane in raport cu un   
 dreptunghi.

9. Proiectii geometrice paralele (definitie, clasificare, exprimarea matriciala - matricea Mort).

10. Proiectii geometrice perspectiva (definitie, clasificare, exprimarea matriciala - matricile Mper si M’per)

11. Determinarea pixelilor interiori ai unui poligon sau ai unei elipse.

12. Modelarea solidelor prin partiționare spațială: arbori quadtrees / octrees (arbori de   
 codificare a ocupării spațiale.

**Ingineria Programării**

Link la fișa disciplinei: <https://profs.info.uaic.ro/~adiftene/Scoala/2016/IP/index.htm>

Subiecte:

1. Etapele necesare dezvoltării aplicațiilor de dimensiuni mari.

2. Modele de dezvoltare a aplicațiilor (XP, Scrum, Agile, TDD).

3. Modelarea aplicațiilor. Diagrame UML, Diagrame de Clase, Diagrame Use-Case.

4. Design pattern. Definiții. GOF, Tipuri de design patterns.

5. Design patterns creaționale. Definiții. Exemple.

6. Design patterns structurale. Definiții. Exemple.

7. Design patterns comportamentale. Definiții. Exemple.

8. Testare software. Definiții. Dilema testării software.

9. Metode de testare. Exemple.

10. Testare manuală.

11. Testare automată.

12. Concepte ale calității programelor.

**Inteligenţă artificială**

Link la fișa disciplinei:

Subiecte:

1.

2.

3.

4.

5.

6.

7.

8.

9.

10.

11.

12.

**Învăţare automată**

Link la fișa disciplinei: https://profs.info.uaic.ro/~ciortuz/Invatare-automata.Licenta.2016f.fisa-disciplinei.RO.pdf

Subiecte:

1. Instante, exemple etichetate, ipoteze. Invatare supervizata (clasificare) vs. invatare nesupervizata (clusterizare). Ipoteze "consistente" cu datele de antrenament (in cazul clasificarii automate).
2. Antrenare, testare, validare. Definitia erorii la antrenare / testare / validare incrucisata / validare incrucisata cu metoda "Leave-One-Out".
3. Overfitting: definitie. Mijloace de prevenire a overfitting-ului (de exemplu, in cazul invatarii arborilor de decizie).
4. Entropie, cistig de informatie: definitii, exemple de utilizare in invatarea automata.
5. Arbori de decizie - definitie. Algoritmul ID3: ideea de baza, caracteristici (adica, ID3 este un algoritm de tip ..., …, ... etc) si extensii.
6. Formula lui Bayes. Functia de verosimiliate si functia de probabilitate a posteriori a datelor. Exemple de algoritmi de invatare automata a. pentru maximizarea verosimilitatii (ML) si b. pentru maximizarea probabilitatii a posteriori (MAP) a datelor.
7. Algoritmii Bayes Naiv si Bayes Corelat: ideea de baza, complexitate (ca numar de parametri de estimat), aplicatii tipice.
8. Algoritmul k-NN: ideea de baza. Clasificare "eager" vs. clasificare "lazy". "Blestemul" marilor dimensiuni. Eroarea asimptotica in raport cu algoritmul Bayes Corelat.
9. Clusterizare ierarhica (top-down vs bottom-up). Tipuri de masuri de similaritate (single- complete- si average-linkage); efecte specifice la clusterizare.
10. Algoritmul K-means: ideea de baza; criteriul "sumei celor mai mici patrate"; problema opririi.
11. Functia de verosimilitate a datelor: definitie. Exemplificarea estimarii parametrilor unor distributii probabiliste in sensul verosimilitatii maxime (MLE): media si varianta distributiei gaussiene uni-variate.
12. Mixtura de distributii gaussiene: formulare generala, exemplificare. Variabile neobservabile. Algoritmul EM pentru mixturi de distributii gaussiene: ideea de baza.

**Limbaje formale, automate şi compilatoare**

Link la fișa disciplinei:

Subiecte:

1.Cum sunt definite gramaticile și limbajele regulate? Ce alte mecanisme de descriere a limbajelor regulate puteți menţiona?

2. Cum se clasifică gramaticile conform ierarhiei lui Chomsky?

3. Ce sunt automatele finite și ce tipuri de automate finite există? Dați un exemplu de automat finit și precizați limbajul acceptat de către acesta.

4. Definiți notiunea de expresie regulata. Dați exemplu de limbaj care nu poate fi descris cu ajutorul unei expresii regulate.

5. Dati 2 exemple de limbajele independente de context. Ce mecanisme de descriere a limbajelor independente de context cunoasteti?

6. Explicați ce legatura exista intre automate finite deterministe si nedeterministe. Care este principala diferența dintre acestea?

7. Cum este definit un arbore de derivare pentru un cuvânt într-o gramatica independenta de context? Dati un exemplu.

8. Ce sunt automatele pushdown si ce clasa de limbaje accepta acestea?

9. Analiza lexicală: definiție, rolul analizorului lexical in cadrul unui compilator.

10. Analiza sintactică: precizați rolul analizorului sintactic in cadrul unui compilator, exemplificați algoritmi pentru analiza sintactica (parsare).

11. Analiza sintactica ascendenta: precizați structura unui parser ascendent general.

12. Precizați diferența dintre metodele de analiza sintactica ascendenta și descendenta.

**Logică pentru informatică**

Link la fișa disciplinei: <https://profs.info.uaic.ro/~masalagiu/l.php>

Subiecte:

1.Conceptul de ***sintaxă*** în Logică. Definiți sintaxa Logicii Propoziționale (**LP**). Dați două exemple de formule din **LP**.

2.Conceptul de ***semantică*** în Logică. Definiți noțiunea de structură (interpretare, asignare) în Logica Propozițională (**LP**). Definiți valoarea de adevăr a unei formule într-o structură. Dați exemplu de o formulă și de o structură și stabiliți valoarea de adevăr a formulei în structura considerată.

3.Conceptul de ***semantică*** în Logică. Definiți noțiunile de formulă satisfiabilă și de formulă validă în Logica Propozițională (**LP**). Dați un exemplu de formulă satisfiabilă și un exemplu de formulă validă (justificare).

4.Conceptul de ***semantică*** în Logică. Definiți noțiunea de formule echivalente în Logica Propozițională (**LP**). Dați exemplu de două formule echivalente și de două formule care nu sunt echivalente (justificare).

5.Legătura dintre ***sintaxă*** și ***semantică*** în Logică (importanța formelor normale). Definiți Forma Normală Conjunctivă (**FNC**) pentru formulele din Logica Propozițională (**LP**). Dați exemplu de o formulă care nu este în **FNC** și calculați o formulă echivalentă cu ea, care este în **FNC**.

6.Conceptul de ***semantică*** în Logică. Definiți noțiunea de consecință semantică în Logica Propozițională (**LP**). Dați exemplu de trei formule F, G, H cu proprietatea că H este consecință semantică din {F, G}.

7.Conceptul de ***sintaxă*** în Logică. Definiți constructiv sintaxa Logicii (cu predicate) de Ordinul I (**LP1**). Dați două exemple de formule din **LP1**.

8.Conceptul de ***semantică*** în Logică. Definiți noțiunea de structură în Logica (cu predicate) de Ordinul I (**LP1**). Dați exemplu de o formulă și de o structură și calculați valoarea de adevăr a formulei în acea structură.

9.Legătura dintre ***sintaxă*** și ***semantică*** în Logică (rezoluție). Care este regula rezoluției în Logica Propozițională (**LP**)? Dați un exemplu de respingere prin rezoluție (justificare).

10.Conceptul de ***sintaxă*** în Logică. Definiți noțiunile de literal și respectiv de clauză în Logica (cu predicate) de Ordinul I (**LP1**). Dați un exemplu de clauză.

11.Conceptul de ***sintaxă*** în Logică. Definiți noțiunile de apariție liberă și respectiv de apariție legată a variabilelor în formulele Logicii (cu predicate) de Ordinul I (**LP1**). Dați exemplu de o formulă și marcați aparițiile libere și respectiv aparițiile legate ale variabilelor în acea formulă.

12.Legătura dintre ***sintaxă*** și ***semantică*** în Logică. Definiți Forma Normală Skolem (**FNS**) pentru formulele Logicii (cu predicate) de Ordinul I (**LP1**) și dați exemplu de o formulă F care nu se află în **FNS**. Găsiți, prin aplicarea lemei de skolemizare, o formulă G aflată în **FNS** astfel încât G să fie slab echivalentă cu F.

**Probabilităţi şi statistică**

Link la fișa disciplinei:

Subiecte:

1.

2.

3.

4.

5.

6.

7.

8.

9.

10.

11.

12.

**Programare avansată**

Link la fișa disciplinei: <https://profs.info.uaic.ro/~acf/java/Fisadisciplinei-Programareavansata.pdf>

Subiecte:

1. Definiti notiunea de *cod de octeti (bytecode)* si modul in care este obtinuta portabilitatea si neutralitatea arhitecturala a aplicatiilor Java.

2. Care sunt modalitatile de alocare, respectiv de eliberare a memoriei pe platforma de programare Java? Care este diferenta dintre zona de memorie *heap* si zona de memorie *stack*?

3. Explicati rolul modificatorului *static* aplicat membrilor unei clase (variabile si metode). Cum este acesta utilizat in implementarea unei clase de tip *Singleton*?

4. Cum sunt declarate si care este rolul interfetelor in limbajul de programare Java? Dati un exemplu de tip referinta avand mai mult de o implementare.

5. Explicati pe scurt arhitectura colectiilor pe platforma Java. Care este diferenta intre colectiile de tip *List* si *Map* si ce implementari avem la dispozitie pentru acestea?

6. Descrieti modalitatile de tratare a exceptiilor in Java. Care este rolul propagarii exceptiilor?

7. Explicati modul in care arhitectura fluxurilor de date Java implementeaza sablonul de proiectare *Decorator*. Dati exemple sugestive de utilizare a fluxurilor de date din acest punct de vedere, atat pe octeti cat si pe caractere.

8. Explicati modul de creare si lansare a unui fir de executie pe platforma Java. Care este utilitatea modificatorului *synchronized* aplicat unei metode?

9. Descrieti pe scurt modul in care poate fi implementata o aplicatie client-server pe platforma Java, folosind protocolul TCP. Ccare sunt clasele necesare si cum vor fi acestea utilizate?

10. Explicati motivul pentru care tehnologia JDBC a fost definita la nivel abstract (prin interfete). Descrieti tipurile de date folosite pentru reprezentarea: a) unei conexiuni cu o baza de date relationala; b) unei comenzi SQL; c) rezultatului obtinut in urma unei interogari SQL.

11. Definiti notiunea de *adnotare* si dati exemple de adnotari din API-ul standard Java. Cum ar putea fi identificate la momentul executiei (runtime) metodele unei clase marcate cu o adnotare avand numele “*Test”*?

12. Definiti termenii de *internationalizare* si *localizare* a aplicatiilor. Care sunt elementele pe care trebuie sa le avem in considerare si care sunt tehnicile de baza pentru crearea unei aplicatii internationalizate?

**Programare orientată-obiect**

Link la fișa disciplinei:

Subiecte:

1. Ce este o functie virtuala pura + C++?

2. Enumerati diferentele si asemanarile intre un pointer si o referinta ?

3. Care este rolul unui template specializat ? Exemple.

4. Principiul lui Liskov.

5. Care este rolul derivarii virtuale in C++ ? Exemple.

6. Design pattern-ul Composite. Exemple de utilizare.

7. Functii friend. Utilizare si limitari. Exemple.

8. Principiul inversiunii dependentelor.

9. Supraincarcarea operatorului “\*”. Exemple de utilizarer.

10. Design pattern-ul Visitor. Exemple de utilizare.

11. For each in C++11. Clase cu suport pentru foreach. Exemple.

12. Design pattern-ul Observer. Exemple de utilizare.

**Proiectarea algoritmilor**

Link la fișa disciplinei:<https://sites.google.com/site/fiicoursepa/>

Subiecte:

1. Noțiunea de problemă computațională. Probleme de decizie. Probleme de optim.
2. Noțiunea de algoritm. Problemă rezolvată de un algoritm.
3. Complexitatea unui algoritm în cazul cel mai nefavorabil.
4. Complexitatea sortării.
5. Complexitatea medie a unui algoritm.
6. (Geometrie Computațională). Algoritmi pentru înfășurătoarea convexă.
7. Algoritmi greedy. Proprietatea de alegere greedy. Exemplu.
8. Programare dinamică. Noțiunea de subproblemă. Principiul de substructură optimă. Exemplu.
9. Clasa NP. Probleme NP-complete. Exemple.
10. Metoda Backtracking.
11. Algoritmi nedeterministi. Exemplu.
12. Probleme nedecidabile. Exemplu.

**Reţele de calculatoare**

Link la fișa disciplinei: <https://profs.info.uaic.ro/~adria/teach/courses/net/AlboaieLenuta_Fisa%20disciplinei_ReteleDeCalculatoare_ro.pdf>

Subiecte:

1. Topologii de retea. Tipuri de reţele de calculatoare.

2. Modelul TCP/IP.

3. Nivelul reţea - caracteristici. Protocolul IP.

4. Retele private.

5. Nivelul transport.

5. Administrarea conexiunii TCP.

6. TCP versus UDP.

7. Modelul client/server. Interfaţa de programare socket BSD.

8. Sistemul numelor de domeniu (DNS).

9. Protocoale la nivel de aplicaţie (e.g. accesul la terminal, transferul de fişiere, poşta electronică)

10. Paradigma RPC (Remote Procedure Call).

11. Modelul peer-to-peer.

12. Reţele wireless.

**Securitatea informaţiei**

Link la fișa disciplinei:https://profs.info.uaic.ro/~fltiplea/

Subiecte:

1.Sisteme Take-Grant in controlul accesului: stare, tranzitie, scurgere de informatii, complexitatea detectiei scurgerii de informatii

2.Sisteme bazate pe matrici in controlul accesului (matricea de control al accesului): stare, tranzitie, scurgere de informatii, complexitatea detectiei scurgerii de informatii

3.Politica de confidentialitate Bell-LaPadula

4.Politica de integritate Biba

5.Poltica Zidul Chinezesc

6.IPsec: modul transport

7.IPsec: modul tunel

8. IPsec: componenta IKE

9.SSL&TLS: componente si prelucrarea datelor

10.SSL&TLS: schimbul de cheie

11 DNSsec.

12.S/MIME si PGP

**Sisteme de operare**

Link la fișa disciplinei:

Subiecte:

1.

2.

3.

4.

5.

6.

7.

8.

9.

10.

11.

12.

**Structuri de date**

Link la fișa disciplinei: https://profs.info.uaic.ro/~sd/admin/SD\_fisa\_disciplinei.pdf

Subiecte:

1. Algoritm. Limbaj algoritmic.

2. Tip de date. Variabilă.

3. Tablouri si structuri.

4. Complexitatea timp a algoritmilor; notatia asimptotica.

5. Complexitatea timp a algoritmilor recursivi. Teorema Master.

6. Liste liniare. Stiva. Coada.

7. Structuri de tip "max/min heap".

8. Structuri de tip "union-find".

9. Sortare.

10. Căutare. Arbori binari de cautare.

11. Arbori de cautare echilibrati.

12. Tabele de dispersie.

**Tehnologii Web**

Link la fișa disciplinei: https://profs.info.uaic.ro/~busaco/teach/courses/web/web-film.html

Subiecte:

1. Argumentați necesitatea validării datelor semi-structurate (XML – *Extensible Markup Language*) și descrieți succint o metodă de validare. Discutați implicațiile asupra redării conținutului HTML (*HyperText Markup Language*) de către client (i.e. *browser*-ul Web).
2. Caracterizați protocolul HTTP (*HypertText Transfer Protocol*) și rolul lui în dezvoltarea aplicațiilor și/sau serviciilor Web.
3. Discutați asemănările și deosebirile dintre stocarea structurată a datelor (i.e. modelul clasic al bazelor de date) și cea semi-structurată – *e.g.*, modelul arborescent oferit de XML (*Extensible Markup Language*).
4. Discutați utilizarea structurilor de date (generice) în ceea ce privește procesările bazate pe DOM (*Document Object Model*). Ce rol are WebIDL (*Web Interface Description Language*) în specificarea acestor structuri de date?
5. Discutați rolul *cookie*-urilor și/sau sesiunilor Web în ceea ce privește autentificarea utilizatorilor unei aplicații Web și/sau a unui serviciu Web.
6. Explicați succint modul de funcționare și arhitectura generală a unui server Web.
7. Descrieți metodele GET și POST oferite de protocolul HTTP (*HyperText Transfer Protocol*) și rolul acestora în dezvoltarea de aplicații Web via paradigma REST (*REpresentational State Transfer*).
8. Precizați semnificația, plus asemănările și deosebirile dintre următoarele perechi de termeni: “server Web” *vs.* “*proxy* Web”, “*cookie*” *vs.* “sesiune Web”, “serviciu Web” *vs.* “aplicație hibridă (*mash-up*) Web”.
9. Descrieți maniera de transfer asincron al datelor facilitat de suita de tehnologii Ajax (*Asynchronous JavaScript And XML*). Specificați câteva dintre utilizările ei tipice.
10. Specificați asemănările și deosebirile dintre interschimbul de date folosind XML (*Extensible Markup Language*) și JSON (*JavaScript Object Notation*).
11. Descrieți arhitectura generică a unei aplicații Web și rolul unui server de aplicații Web în realizarea acesteia.
12. Definiți ce reprezintă expresia “disponibilitatea serviciilor” și relația acesteia cu tipul de atac “refuz al serviciilor”. Cum are loc un atac DDoS (*Distributed Denial of Service*)?