

Subiecte la proba scrisă a examenului de licență iunie 2014

Cunoștințe fundamentale – teorie

T1. Descrieți metoda de programare Greedy. Propuneți un algoritm bazat pe această metodă pentru rezolvarea problemei continue (fracționare) a rucsacului (descriere, pseudocod, justificarea corectitudinii și complexității algoritmului propus). Dați un exemplu de date de intrare pentru care algoritmul Greedy descris nu furnizează soluția optimă pentru problema discretă a rucsacului.

T2. Să se enunțe și să se demonstreze lema de pompare pentru limbajele independente de context.

T3. Tratați următoarele cerințe:

- (a) Descrieți funcționalitatea celor 5 clase de module care intervin în structura unui SGBD. Analizați în mod special programele de gestiune a bazei de date (PGBD).
- (b) Precizați două tipuri de obiecte ale unei baze de date Oracle care permit optimizarea regăsirii informației. Pentru fiecare tip de obiect indicat, descrieți modul în care acesta intervine în optimizare, dați un exemplu de creare și specificați o interogare pe care obiectul creat o optimizează.
- (c) Formele normale Boyce-Codd (BCNF) și FN4. Comentați, exemplificați, justificați.

T4. Tratați următoarele cerințe:

- (a) Enunțați și demonstrați teorema deducției în calculul propozițional clasic.
- (b) Demonstrați sintactic, folosind teorema deducției, că următoarea formulă este teoremă a calculului propozițional clasic: $\vdash (\psi \rightarrow \alpha) \rightarrow ((\alpha \rightarrow \beta) \rightarrow (\psi \rightarrow \beta))$.

Cunoștințe fundamentale – probleme

P1. Se dau doi vectori, A de dimensiune n și B de dimensiune m, ambii cu componentele ordonate crescător. Se cere listarea valorilor comune celor doi vectori. Să se descrie un algoritm de cost minim pentru fiecare dintre următoarele două situații:

- (a) $n = m$.
- (b) $n \ll m$ (n mult mai mic decât m).

Se vor descrie informal algoritmii, se vor implementa într-un limbaj de programare, se va preciza și justifica complexitatea obținută.

P2. Fie $L = \{\leq\}$ un limbaj de ordinul I, unde \leq este un simbol de relație de aritate 2.

- (a) Determinați o mulțime de formule Γ astfel încât modelele lui Γ să fie mulțimile total ordonate.
- (b) Scrieți o formulă ψ astfel încât următoarea echivalență să fie adevărată, oricare ar fi (A, \leq) o mulțime total ordonată: $A \models \psi$ dacă și numai dacă A are cel mai mic element.
- (c) Demonstrați că enunțul $\forall x \forall y \exists z ((x \leq z) \vee (y \leq z))$ este adevărat în orice mulțime total ordonată.

P3. Se dă un număr $x \leq 100000$. Calculați n - numărul minim de numere prime p_1, p_2, \dots, p_n - astfel încât $p_1 + p_2 + \dots + p_n = x$. Numerele prime nu trebuie să fie distincte. Se va descrie informal algoritmul, se va implementa într-un limbaj de programare, se va preciza și justifica complexitatea obținută.

P4. Tratați următoarele cerințe:

- Să se proiecteze diagrama conceptuală a modelului de date corespunzător următorului scenariu:
Studentii unei facultăți participă la stagii oferite de către companii. Stagiile pot fi de tip dezvoltare sau suport tehnic. Fiecare stagiou are o dată de început și o dată de final. Un stagiou de tip dezvoltare necesită cunoașterea unor limbaje de programare. Pe perioada efectuării unui stagiou studenții pot lucra pe sarcini asociate unor proiecte ale companiei și sunt coordonați de către specialiști din compania respectivă. În timpul unui stagiou, un student lucrează cu anumite tehnologii (limbaje, sisteme).
- Să se creeze structura unui tabel asociativ din diagrama conceptuală de la punctul anterior. Să se adauge constrângerile de integritate asupra acestui tabel.
- Pentru studenții care au participat la cele mai multe stagii, să se obțină numele acestora și numărul total de luni de stagiou pe care le-au efectuat.
- Să se obțină numele specialiștilor care au coordonat doar studenți care au lucrat cu sistemul Oracle pe perioada stagiului.

P5. Se dau 3 șiruri de numere naturale: a de lungime m , b de lungime n și v de lungime $m+n$ ($m, n < 500$). Spunem că șirul v se obține prin interclasarea șirurilor a și b dacă:

- v are ca elemente toate elementele din a și b ;
- elementele din v care provin din a păstrează ordinea în care acestea se găseau în a (dar nu apar neapărat pe poziții consecutive în v);
- elementele din v care provin din b păstrează ordinea în care acestea se găseau în b (dar nu apar neapărat pe poziții consecutive în v).

Atunci șirului v îi putem asocia (nu neapărat în mod unic) un șir binar $s = (s_1, s_2, \dots, s_{m+n})$ cu semnificația:

$$s_i = \begin{cases} 0, & \text{dacă al } i\text{-lea element din } v \text{ provine din șirul } a \\ 1, & \text{dacă al } i\text{-lea element din } v \text{ provine din șirul } b \end{cases}$$

Să se determine în mod eficient ($O(nm)$) dacă v se poate obține prin interclasarea șirurilor a și b . Se va afișa un mesaj corespunzător (da/nu). În caz afirmativ se va afișa și șirul s asociat lui v . Se va descrie informal algoritmul, se va implementa într-un limbaj de programare, se va preciza și justifica complexitatea obținută.

Spre exemplu, pentru $m=3$, $a=(3,1,2)$, $n=4$, $b=(3,1,5,4)$, $v=(3,1,3,5,1,2,4)$ se va afișa „da” și șirul 1101001, iar pentru $m=3$, $a=(3,1,2)$, $n=4$, $b=(3,1,5,4)$, $v=(3,1,2,5,4,3,1)$ se va afișa „nu”.

P6. Problema filozofilor. Enunț și implementare într-un limbaj de programare.

P7. Se dă un alfabet compus din n litere, a_1, a_2, \dots, a_n , cu frecvențele lor de apariție p_1, p_2, \dots, p_n . Se cere să se construiască un cod binar cu proprietatea prefix peste acest alfabet, cod care să minimizeze lungimea medie a mesajelor codificate.

P8. Tratați următoarele cerințe:

- (a) Spuneți dacă limbajul următor este limbaj independent de context sau nu; dacă da, construiți o gramatică independentă de context care să îl genereze, dacă nu, demonstrați că nu este (folosind, spre exemplu, lema de pompare):

$$L_1 = \{a^{n+3}b^{2m} \mid n \neq m\}$$

- (b) Spuneți dacă limbajul următor este limbaj independent de context sau nu; dacă da, construiți o gramatică independentă de context care să îl genereze, dacă nu, demonstrați că nu este (folosind, spre exemplu, lema de pompare):

$$L_2 = \{a^m b^n \mid n > 0, m \text{ este un număr prim}\}$$

Cunoștințe de specialitate

S1. Linii euleriene. Definiții, exemple. Teorema lui Euler (enunț și demonstrație).

S2a. Scrieți algoritmul de propagare înapoi pentru o rețea de perceptroni feed forward cu un singur strat ascuns.

S2b. Jocuri de două persoane cu informație completă:

- (a) Algoritmul MiniMax - pașii algoritmului.
- (b) Algoritmul MiniMax - implementare în Prolog (cu explicarea predicatelor care intervin).
- (c) Algoritmul Alpha-Beta: prezentare generală cu exemplu explicat

S3. Gramatici și limbaje de tip LL(1) (definiții, caracterizări echivalente). Algoritmul de analiză sintactică 1-predictiv.

S4. Criptanaliza sistemelor de criptare. Definiție. Minim 2 tipuri de atac (descriere, exemple de sisteme vulnerabile, modalități de prevenire).

S5. Echivalența variantelor de mașini Turing.

S6. Fire de execuție:

- (a) Creare.
- (b) Modelul aleator; un exemplu ilustrativ.
- (c) Excluderea reciprocă; un exemplu ilustrativ.
- (d) Un mod de realizare a excluderii reciproce în Java; aplicație la unul dintre exemplele precedente.

S7. Tratați următoarele cerințe:

- (a) Variabile const în C: sintaxă, particularități. Câmpuri de date const: sintaxă, particularități. Metode const în C++: sintaxă, particularități, scop, exemplu.
- (b) Încapsularea (definirea claselor) în C++: scop, definiție, sintaxă, specificatori de acces, exemplu.
- (c) Polimorfismul de execuție în C++: scop, definiție, sintaxă, mecanism de implementare, exemplu.

S8. Fie (S, Σ) o semnătură multisortată și X o mulțime multisortată de variabile:

- (a) Definiți algebra de termeni $T_{\Sigma}(X)$.
- (b) Demonstrați că $T_{\Sigma}(X)$ este (S, Σ) -algebră inițială.
- (c) Explicați rolul algebrei $T_{\Sigma}(X)$ în programarea bazată pe logica ecuațională multisortată.