

Universitatea din București
Facultatea de Matematică-
Informatică

Nume: _____

Calculabilitate și Complexitate

Puncte: 26. Timp: 52 min

Data: 7-02-2023

Examen, 7 Februarie, Nivelul I,

Subiecte: **A**

Instrucțiuni I. Incercuiți răspunsul/răspunsurile corecte. Toate întrebările contează în mod egal. Puteți folosi marginile hârtiei drept ciornă, **dar niciun alt material**. O întrebare poate avea mai multe răspunsuri corecte. Toate contează în mod egal. **Pe de altă parte, dacă alegeți un răspuns greșit, punctajul vostru la întrebare este zero.**

1. Funcția lui Ackermann **nu** este

- (a) primitiv recursivă.
- (b) recursivă.
- (c) parțial recursivă.
- (d) niciunul dintre celelalte răspunsuri nu este corect.

2. Care operație **nu** este folosită în definiția unei funcții primitiv recursive ?

- (a) compunere
- (b) recursie primitivă
- (c) minimizare
- (d) exponențiere

3. O funcție parțial recursivă este recursivă doar dacă ...

- (a) este obținută prin minimizare.
- (b) este definită în orice punct x .
- (c) poate fi calculată de o mașină Turing.
- (d) crește mai rapid decât orice funcție primitiv recursivă.

4. Cum putem crea o funcție care **nu** e primitiv recursivă ?

- (a) enumerăm toate funcțiile primitiv recursive. Creăm o funcție care pe inputul i returnează valoarea $f_i(i) + 1$.
- (b) folosim operația de minimizare, dacă rezultatul nu este definită pentru toate inputurile i .
- (c) prin compunere din două funcții primitiv recursive.
- (d) Cu un automat finit.

5. Dacă A este o mulțime recursiv enumerabilă iar K este problema opririi, atunci
- (a) $A \leq_m K$.
 - (b) $K \leq_m A$.
 - (c) Ambele reduceri sunt adevărate.
 - (d) Nicio reducere nu e adevărată.
6. Dacă A, B sunt probleme de decizie iar $A \oplus B = \{x0|x \in A\} \cup \{y1|y \in B\}$ atunci
- (a) $A \leq_m A \oplus B$.
 - (b) $A \oplus B \leq_m A$.
 - (c) Ambele reduceri sunt adevărate.
 - (d) Nicio reducere nu e adevărată
7. Care din următoarele de calcul recunoaște cea mai largă clasă de limbaje ?
- (a) automat finit
 - (b) mașină Turing cu o bandă.
 - (c) mașină Turing cu două benzi.
 - (d) Toate modelele menționate recunosc aceeași clasă de limbaje.
8. O mașină Turing universală ...
- (a) poate simula orice mașină Turing.
 - (b) acceptă un limbaj recursiv enumerabil.
 - (c) are o bandă specială, numită *bandă oracol*.
 - (d) se oprește intotdeauna într-un număr finit de pași.
9. Care din problemele următoare *nu* sunt recursive ?
- (a) problema opririi K .
 - (b) problema de a decide dacă un număr este prim.
 - (c) problema satisfiabilității.
 - (d) Niciuna din problemele listate.
10. Care din clasele următoare de probleme de decizie sunt închise la operația de complementare (dacă $A \in \mathcal{C}$ atunci $\bar{A} \in \mathcal{C}$) ?
- (a) clasa problemelor recursive
 - (b) clasa problemelor primitiv recursive.
 - (c) clasa problemelor parțial recursive.
 - (d) P clasa problemelor care au algoritmi polinomiali.

11. Care din problemele următoare **nu** sunt recursive ?
- (a) $K_1 = \{ \langle x, y \rangle : M_x(y) \text{ se oprește într-un pas} \}$
 - (b) $K = \{ \langle x, y \rangle : M_x(y) \text{ se oprește} \}$.
 - (c) $\overline{K} = \{ \langle x, y \rangle : M_x(y) \text{ nu se oprește} \}$
 - (d) toate problemele sunt recursive.
12. Care din următoarele probleme sunt decidabile ?
- (a) Fiind date o mulțime de tipuri de pavaje, problema dacă putem pava planul cu pavajele Wang date.
 - (b) Fiind date o mulțime de tipuri de pavaje, problema dacă putem pava un pătrat 1000x1000 cu pavajele Wang date
 - (c) Fiind dat un polinom cu coeficienți întregi $p(x_1, x_2, \dots, x_n)$ există soluții întregi pentru ecuația $p(x_1, x_2, \dots, x_n) = 0$?
 - (d) Fiind dat un polinom cu coeficienți întregi $p(x_1, x_2, \dots, x_n)$ există soluții pentru ecuația $p(x_1, x_2, \dots, x_n) = 0$ cu $|x_1|, \dots, |x_n| \leq 1000$?
13. Un exemplu de funcție care *nu* este parțial recursivă este:
- (a) Funcția lui Ackermann
 - (b) Funcția "Busy Beaver"
 - (c) Funcția $f(x)$ care numără pașii făcuți de o mașină Turing universală M_U pe intrarea x (nedefinită dacă M_U nu se oprește)
 - (d) Toate funcțiile de mai sus sunt parțial recursive.
14. Fiind dată o mașină Turing deterministă, care din afirmațiile următoare sunt adevărate ?
- (a) Dacă pe o intrare x mașina rulează în $f(|x|)$ pași, atunci pe orice intrare y spațiul folosit de mașină este $O(f(|y|))$.
 - (b) Dacă pe o intrare x mașina rulează în spațiu $f(|x|)$, atunci pe orice intrare y mașina rulează în $O(f(|y|))$ pași.
 - (c) Dacă pe o intrare x mașina nu se oprește atunci spațiul folosit de $M(x)$ este infinit.
 - (d) niciuna din afirmații nu este adevărată.
15. Fiind dată formula următoare: $x \vee y \vee z, \bar{y} \vee \bar{t}, \bar{z} \vee \bar{t}$, care literali sunt puri ?
- (a) x
 - (b) y
 - (c) z
 - (d) t

16. Care din următoarele afirmații sunt adevărate ?
- (a) Orice algoritm polinomial nedeterminist poate fi simulat de un algoritm determinist cu complexitate polinomială.
 - (b) Orice algoritm polinomial nedeterminist poate fi simulat de un algoritm determinist cu complexitate $O(2^{n^{O(1)}})$.
 - (c) Dându-se o formulă booleană în care toate variabilele sunt cuantificate, există un algoritm polinomial pentru a decide dacă formulele sunt adevărate sau false.
 - (d) Pentru o parte a răspunsurilor de mai sus nu se cunoaște statutul lor de adevăr.
17. Care din următoarele probleme **nu** sunt cunoscute ca fiind NP-complete ?
- (a) 2-SAT
 - (b) 3-SAT
 - (c) Problema de a decide dacă un număr este prim sau nu.
 - (d) problema Vertex Cover.
18. Care din următoarele afirmații sunt adevărate ?
- (a) Dacă $A \leq_m^P B$ și $B \in P$ atunci $A \in P$.
 - (b) Dacă $A \leq_m^P B$ și $B \in NP$ atunci $A \in NP$.
 - (c) Dacă $A \leq_m^P B$ și B este NP-completă atunci A este NP-completă.
 - (d) Dacă $A \leq_m^P B$ și B este NP-hard atunci A este NP-hard.
19. Care din următoarele afirmații este adevărată ?
- (a) O formulă logică CNF cu cel mult doi literali în fiecare clauză are demonstrații prin rezoluție de lungime polinomială în n .
 - (b) O formulă logică CNF cu cel mult trei literali în fiecare clauză are demonstrații prin rezoluție de lungime polinomială în n .
 - (c) Există un algoritm de tip Davis-Putnam care pe instanțe nesatisfiabile pentru 2-SAT rulează în timp polinomial în n .
 - (d) Există un algoritm de tip Davis-Putnam care pe instanțe nesatisfiabile pentru 3-SAT rulează în timp polinomial în n .
20. Care din următoarele probleme au algoritmi polinomiali ?
- (a) Fiind dată o mulțime de numere, sunt ele în ordine sortată ?
 - (b) Fiind dat un graf orientat G și două vârfuri s, t , putem ajunge de la s la t ?
 - (c) Putem colora un graf cu 2 culori astfel încât orice două vârfuri adiacente să aibă culori diferite ?
 - (d) Fiind dată o formulă propozițională în forma normală conjunctivă în care în fiecare clauză apare cel mult un literal pozitiv, este formula satisfiabilă ?

21. Dacă $P = NP$ atunci ...
- (a) Putem colora un graf cu numărul minim de culori în timp polinomial.
 - (b) putem rezolva orice problemă cu un algoritm polinomial.
 - (c) putem găsi o soluție pentru problema Vertex Cover cu un algoritm cu complexitate polinomială.
 - (d) Orice problemă rezolvabilă în timp polinomial folosind SAT ca subrutină are un algoritm polinomial.
22. Care din următoarele clase de complexitate are probleme complete ?
- (a) $co-NP$.
 - (b) toate clasele Σ_k^P din ierarhia polinomială.
 - (c) PSPACE.
 - (d) Niciuna din clase.
23. Teorema lui Cook-Levin ...
- (a) Arată faptul că SAT este NP-completă.
 - (b) Simulează funcționarea unei Mașini Turing nedeterministe polinomiale cu o formulă logică.
 - (c) Dă un algoritm polinomial pentru SAT.
 - (d) Arată că nu există un algoritm polinomial pentru SAT.
24. Dacă $P = NP$ atunci ...
- (a) $NP = co - NP$.
 - (b) Putem găsi în timp polinomial un martor pentru o instanță pozitivă a unei probleme din NP.
 - (c) Orice problemă NP-hard este în P
 - (d) Putem găsi o clică maximală într-un graf în timp polinomial.
25. Un algoritm nedeterminist care folosește spațiu $s(n)$ poate fi simulat de ...
- (a) un algoritm determinist care folosește spațiu $O(s^2(n))$.
 - (b) Un algoritm determinist polinomial.
 - (c) Un algoritm pentru problema opririi.
 - (d) Niciun răspuns nu e corect.

26. Care probleme NP-complete pot fi folosite in practică pentru rezolvarea altor probleme NP-complete ?

(a) SAT

(b) ILP, problema programării liniare in numere intregi.

(c) problema izomorfismului a două grafuri

(d) niciuna, nu există algoritmi practici pentru probleme NP-complete.

Answer Key for Exam A

Instrucțiuni I. Incercuiți răspunsul/răspunsurile corecte. Toate întrebările contează în mod egal. Puteți folosi marginile hârtiei drept ciornă, **dar nicun alt material**. O întrebare poate avea mai multe răspunsuri corecte. Toate contează în mod egal. **Pe de altă parte, dacă alegeți un răspuns greșit, punctajul vostru la întrebare este zero.**

1. (a)
2. (c), (d)
3. (b)
4. (a), (b)
5. (a)
6. (a)
7. (b), (c)
8. (a), (b)
9. (a)
10. (a), (b) , (d)
11. (b), (c)
12. (b), (d)
13. (b)
14. (a), (d)
15. (a), (d)
16. (b), (d)
17. (a), (c)
18. (a), (b)
19. (a), (c)
20. (a), (b) , (c) , (d)
21. (a), (c)

22. (a), (b) , (c)

23. (a), (b)

24. (a), (b) , (d)

25. (a)

26. (a), (b)