Examen la Analiza Algoritmilor
Set 2 – 04/02/2018

Timp de rezolvare: 90 de minute

1. (1,5p) Care dintre următoarele probleme este în NP-hard, dar nu în NP:
    a) Ciclu hamiltonian; b) Problema opririi; c) SAT
2. (1,5p) Putem găsi o reducere de la problema corespondențelor lui Post (PCP) la SAT?
    a) Da, o reducere Turing; b) Da, o reducere polinomială; c) Nu, acest lucru este imposibil
3. (1,5p) Fie f(n) ∈θ(n**3) și g(n)∈Ω(n**2), atunci:
    a) f(n)/g(n)∈θ(n); b) f(n)/g(n)∈O(n); c) f(n)/g(n)∈Ω(n)
4. (1,5p) Problema determinării dacă două vârfuri dintr-un graf orientat fac parte din aceeași componentă tare conexă este (alegeți clasa de complexitate cea mai restrictivă):
    a) NLOGSPACE; b) P; c) decidabilă
5. (1,5p) Care este complexitatea amortizată pentru o operație de inserție într-un tabel (vector) dinamic care își dublează spațiul atunci când este plin:
    a) θ(n); b) θ(log n); c) θ(1)

6. (4p) Demonstrați teorema lui Savitch: Dacă oproblemă A∈NSPACE(f(n)), atunci A∈DSPACE(f(n)**2) ∀ f(n)∈Ω(log n). Dacă nu știți demonstrația în cazul general, demonstați faptul că acest lucru este valabil pentru problema GAP (punctaj parțial).

7. (4p) Demonstrați decidabilitatea următoarei probleme: Primind la intrare un program P și o mulțime finită A, determinați dacă P(x) se termină pentru toate elementele x∈A.

8. (3p) a) Dați un exemplu de graf neorientat conex cu 7 vârfuri care are o acoperire minimală cu vârfuri de dimensiunea 2.
   (3.5p) b) Descrieți funcția de transformare a datelor de intrare pentru reducerea polinomială 3-SAT $\leq_P$ k-acoperire.

9. (5p) Scrieţi un algoritm nedeterminist pentru următoarea problemă:
Fie o muțime S de numere reale și doi întregi pozitivi, k și l. Este posibilă partiționarea lui S în k submulțimi disjuncte astfel încât fiecare submulțime să conțină 2l elemente, iar suma elementelor din fiecare submulțime să fie zero?

10. (3p) Verificaţi posibilitatea aplicării teoremei master şi calculaţi limite asimptotice de complexitate pentru recurenţa: T(n) = 32 T(n/2) + (log (n**n))**5

11. (10p) Fie tipul de date TLIST, lista generica cu elemente de tip T, definita prin constructorii:
    [] : -> TLIST
    [a] : T -> TLIST
    cons(e, l) : T * TLIST - > TLIST // cons(e, l) reprezintă adăugarea elementului e la începutul listei l

Se cunosc operatorii:

```
member: TLIST → Bool
(M1)  member(b, []) = false
(M2)  member(b, [a]) = a==b
(M3)  member(b, cons(e, l)) = (b==e) || member(b, l)
nodup: TLIST → BOOL  //verifica faptul ca o lista nu contine elemente duplicate
(N1)  nodup([]) = true
(N2)  nodup([a]) = true
(N3)  nodup(cons(e, l)) = (member(e,l)==false) && nodup(l)
union: TLIST * TLIST → TLIST  // similar cu reuniunea pe mulţimi, are grijă să nu introducă duplicate
(U1)  union([], l2) = l2
(U2)  union([a], l2) = member(a, l2)? l2:cons(a,l2)
(U3)  union(cons(e, l), l2) = member(e, l2)? union(l, l2):cons(e, union(l,l2))
```

Să se demonstreze prin inducție structurală proprietatea:

```
nodup(l1) && nodup(l2) -> size(union(l1, l2)) >= size(l1) ∀ l1,l2 ∈ TLIST
```

Total: 40p