Examen la Analiza Algoritmilor
04/02/2018 - Set 1

Timp de rezolvare: 90 de minute

1. (1,5p) Știm că $A \leq_T B$, $F:I_A \rightarrow I_B$ funcția de transformare a a datelor de intrare, iar I_A este mulțimea vectorilor. Dacă în cadrul funcției F operația cea mai costisitoare este construirea tuturor tuplurilor de k (k-fixat, de ex. k=3) elemente din $v[1..n] \in I_A$, este aceasta o reducere polinomială corectă?
   a) Da; b) Nu; c) Depinde de cele două probleme implicate în reducere
2. (1,5p) Problema determinării dacă un program P, primind la intrare x, se termină în mai puțin de un an este:
   a) nedecidabilă; b) semidecidabilă; c) decidabilă
3. (1,5p) Dacă o problemă este în clasa de complexitate NTIME(f(n)), atunci ea este sigur și în clasa:
   a) PTIME(f(n)); b) NSPACE(f(n)); c) PSPACE((log n)**2)
4. (1,5p) Care este complexitatea algoritmului de sortare rapidă (quicksort) în cazul cel mai defavorabil, dacă nu se folosește alegerea aleatoare a pivotului:
   a) Θ(n log n); Θ(n**2); c) Θ(n)
5. (1,5p) Știind că $A \leq_P SAT$, ce putem spune sigur despre problema A:
   a) A∈P ; b) A∈NP-complete; c) A este decidabilă

6. (4p) Demonstrați prin metoda potențialului că operația de inserare a unui element într-un tablou (vector) dinamic care își dublează dimensiunea când este plin are o complexitate amortizată Θ(1).

7. (4p) Pornind de la enunțul problemei k-acoperire cu vârfuri, prezentați doi algoritmi de aproximare pentru aceasta, dintre care cel puțin unul să aibă un factor de aproximare constant.

8. (3p) a) Dați un exemplu de formulă 3-FNC cu maxim 4 variabile care să nu fie satisfiabilă. Justificați pe scurt.
   (3.5p) b) Schițați schema de reducere polinomială de la problema SAT la problema 3-SAT.

9. (5p) Scrieţi un algoritm nedeterminist polinomial pentru următoarea problemă și determinați complexitatea sa: Pentru startup-ul vostru, FaceAlgo, doriți să atribuiți proiectele angajaților știind că fiecare angajat poate lucra la un singur proiect la un moment dat. Se dau setul P de proiecte și E de angajați, iar pentru fiecare proiect p∈P o submulțime Ep⊆E de angajați care trebuie alocați proiectului p pentru a-l finaliza în timp util. Determinați dacă se pot aloca angajații pe proiecte astfel încât să fie finalizate (cel puțin) k proiecte.

10. (3p) Verificaţi posibilitatea aplicării teoremei master şi calculaţi limite asimptotice de complexitate pentru recurenţa: T(n) = 3T(n/8) + Θ(n**1/3).

11. (10p) Fie tipul de date TLIST, o listă generică cu elemente de tip T, definită prin constructorii:
   [] : -> TLIST
   [a] : T -> TLIST
   cons(e, l) : T * TLIST - > TLIST

Se cunosc, de asemenea, operatorii definiți prin axiomele următoare:

```
head : TLIST → TLIST
(H1) head([]) = []
(H2) head([a]) = [a]
(H3) head(cons(e, l)) = [e]

tail : TLIST → TLIST
(T1) tail([]) = []
(T2) tail ([a]) = []
(T3) tail (cons(e, l)) = l
```

```
equal : TLIST x TLIST → BOOL
(E1) equal([], l2) = l2==[]
(E2) equal([a], l2) = l2==[a]
(E3) equal(cons(e, l), l2) = ([e]==head(l2))
                             && equal(l, tail(l2))
map: Fun * TLIST → TLIST (Fun reprezinta
multimea functiilor f:T->T)
(M1)  map(f, []) = []
(M2)  map(f, [a]) = [f(a)]
(M3)  map(f, cons(e, l)) = cons(f(e), map(f,l))
```

Fiind dat faptul că `equal(l1,l2) -> l1==l2, ∀ l1,l2 ∈ TLIST`, să se demonstreze prin inducție structurală proprietatea:

`equal(l1, l2) -> equal(map(f, l1), map(f,l2)), ∀ l1,l2 ∈ TLIST`

Total: 40p