1. Fie tipul de date BTree definit prin constructorii:

BTempty: -> BTree
BTnode: BTree x T x BTree -> BTree

si definitiile:

flattenTree(t) : BTree -> List
(F1) flattenTree(BTempty) = []
(F2) flattenTree(BTnode(left, i, right)) = flattenTree(left) ++ [i] ++ flattenTree(right)

numBTelem(t): BTree-> N
(N1) numBTelem(BTempty) = 0
(N2) numBTelem(BTnode(left, i, right)) = 1 + numBTelem(left) + numBTelem(right)

length(l) : LIST -> N
(L1) length([]) = 0
(L2) length(h:t) = 1 + length(t)

Stiind ca avem constructorii [], [a] si h:t (echivalent cu cons(h,t)) pentru tipul LIST si ca operatorul ++ (concatenarea a doua liste) este definit astfel incat urmatoarea proprietate este adevarata pentru orice doua liste l1, l2: length(l1 ++ l2) = length(l1) + length(l2), sa se demonstreze ca:

numBTelem(t) = length (flattenTree(t)) pentru orice t de tip BTree

2. O expresie aritmetica complet parantezata este definita astfel:

0, 1, x, [e1+e2], [e1*e2], [-e2]
2 constructori nulari, 1 constructor extern, 3 constructori interni. Sa notam cu E acest tip de date.

Se definesc operatorii:
eval(e, n): E x N -> N
(E1) eval(0, n) = 0
(E2) eval(1, n) = 1
(E3) eval(x, n) = n
(E4) eval([e1+e2], n) = eval(e1, n) + eval(e2, n)
(E5) eval([e1*e2], n) = eval(e1, n) * eval(e2, n)
(E6) eval([-e1], n) = -eval(e1, n)

subst(e, f): E x E -> E
(S1) subst(0, f) = 0
(S2) subst(1, f) = 1
(S3) subst(x, f) = f
(S4) subst([e1+e2], f) = [subst(e1,f) + subst(e2, f)]
(S5) subst([e1*e2], f) = [subst(e1,f) * subst(e2, f)]
(S6) subst([-e], f) = [-subst(e, f)]

Sa se demonstreze prin inductie structurala ca pentru orice e, f din E si n din N, proprietatea urmatoare este adevarata:

eval(subst(e, f), n)= eval(e, eval(f, n))

3. Demonstrati ca problema 2-SAT este in P.

Problema 2-SAT este un caz particular al SAT: Fie o formula F in FNC (forma normala conjunctiva) in care fiecare termen contine 2 variabile. Este F satisfabila ?
Indicatie: Gasiti o reducere a problemei 2-SAT la o problema din P.

4. Problema FND-tautologie este in P sau NP ? Problema FNC-tautologie este in P sau NP ? Demonstratie si discutie.

O formula in forma normal-disjunctiva (FND) este alcatuita dintr-o disjunctie de conjuctii de literali (variabile sau variabile negate).
O formula in forma normal-conjunctiva (FNC) este alcatuita dintr-o conjunctie de disjunctii de literali.
O formula logica F este o tautologie daca F este adevarata pentru orice asignare a variabilelor ce o alcatuiesc.

5. Construiti un algoritm nedeterminist pentru rezolvarea problemei colorarii unui graf cu k culori. Care este complexitatea acestui algoritm ? Din ce clasa face parte aceasta problema (scurta justificare) ?

Problema colorarii unui graf cu k culori (decizie): Se dau un graf G(V, E) si k culori distincte. Se pot colora nodurile grafului folosind cele k culori astfel incat pentru orice muchie (u, v) a grafului, culoare(u) != culoare(v) ?

6. Construiti un algoritm determinist (cat mai bun) pentru rezolvarea problemei anterioare. Care este complexitatea acestui algoritm ?

7. Aratati ca problema 0/1 programare intreaga este NP-grea.

*0/1 Programare intreaga: Se da o matrice A(n x n) si un vector b(n). Exista un vector x cu n elemente din multimea {0, 1} astfel incat A * x >= b ?*

8. Stabiliti o reducere polinomiala intre problema sumelor (q-sume) si problema rucsacului. Discutie referitor la clasa acestor probleme.

Problema rucsacului: Se dau n obiecte ce au valoarea v[i] si greutatea w[i] si un rucsac in care se poate transporta o greutate maxima W. Putem transporta cu rucsacul obiecte care sa aiba cel putin valorea V fara a depasi greutatea W ? (Observatii: 1) un obiect se ia in rucsac ca intreg, nu fractionar; 2) pentru reducerea polinomiala, puteti introduce si alte constrangeri - de exemplu, limitarea valorii transportate in rucsac, etc.)

9. Sa presupunem ca am gasit un algoritm de rezolvare pentru problema k-clica care este determinist si polinomial. Ce concluzii puteti trage din aceasta descoperire ?

Notare: Fiecare exercitiu valoreaza 1p => 9 * 1p + 1p (oficiu) = 10p