

Temă pentru acasă - partea C.

14 puncte

C1. (3 puncte - Las Vegas) Avem un *game tree* în care fiecare frunză se găsește la aceeași distanță $2h$ față de rădăcină și conține o valoare booleană, iar fiecare nod intern (inclusiv rădăcina) are exact trei descendenți direcți (copii). Fiecare nod intern aflat la o distanță pară față de rădăcină este etichetat cu AND iar fiecare nod intern aflat la o distanță pară față de rădăcină este etichetat cu OR. Implementați în R un algoritm pentru evaluarea acestui tip de arbore (determinarea valorii booleene din rădăcină).

Indicație: cursul 10.

C2. (6 puncte - Las Vegas) Problema mariajului stabil încearcă determinarea unei așa numite stabilități între două mulțimi disjuncte de același cardinal: $\mathcal{M} = \{M_1, M_2, \dots, M_n\}$ (bărbați) și $\mathcal{W} = \{W_1, W_2, \dots, W_n\}$ (femei); un *cuplaj* este o bijecție $f : \mathcal{M} \rightarrow \mathcal{W}$. Fiecare persoană are o lista de preferințe a membrilor celeilalte mulțimi (listele sunt ordonate în ordinea inversă a atractivității). Un cuplaj este *f instabil* dacă există $M_i, M_j \in \mathcal{M}$ astfel încât M_i este mai mult atras de $f(M_j)$ decât de $f(M_i)$ iar $f(M_j)$ atras mai mult de M_i decât de M_j . Un cuplaj care nu este instabil este numit *stabil*.

Implementați următorul algoritm aleator pentru determinarea unui cuplaj stabil.

```
for ( $j = \overline{1, n}$ ) do
   $f(M_i) = -1$ ;  $f^{-1}(W_i) = -1$ ; // inițial cuplajul nu este definit.
end for
while ( $\exists M_i$  astfel ca  $f(M_i) = -1$ ) do
  fie  $M_i$  astfel ca  $f(M_i) = -1$ ;
  alege aleator și uniform un  $j \in \{1, 2, \dots, n\}$  iar  $M_i$  face o "propunere" către  $W_j$ ;
  if ( $f^{-1}(W_j) \neq -1$  and  $W_j$  preferă pe  $M_i$  lui  $M_h = f^{-1}(W_j)$ ) then
     $f(M_h) = -1$ ;  $f(M_i) = W_j$ ;  $f^{-1}(W_j) = M_i$ ;
  end if
  if ( $f^{-1}(W_j) == -1$ ) then
     $f(M_i) = W_j$ ;  $f^{-1}(W_j) = M_i$ ;
  end if
end while
```

C3. (5 puncte - Monte Carlo) pentru un cuvânt $v = v_1 v_2 \dots v_n \in \{0, 1\}^n$, definim

$$number(v) = \sum_{i=1}^n v_i 2^{i-1}.$$

Să presupunem că dorim să verificăm dacă un cuvânt dat $u \in \{0, 1\}^n$ aparține unei mulțimi de cuvinte $U = \{u^1, u^2, \dots, u^m\}$. Implementați următorul algoritm aleator pentru rezolvarea acestei probleme.

```
uniformly choose a prime number  $p \leq n^2$ ; // folosiți funcția Primes din pachetul numbers.
 $r = number(u) \pmod{p}$ ;
for ( $j = \overline{1, m}$ ) do
   $r_j = number(u^j) \pmod{p}$ ;
end for
if ( $r \in \{r_1, r_2, \dots, r_m\}$ ) then
  return "u ∈ U";
end if
return "u ∉ U";
```

Rezolvările acestor exerciții (funcțiile R și apelurile lor) vor fi redactate într-un singur script R.