Temă pentru acasă - partea C.

14 puncte

C1. (**3 puncte - Las Vegas**) Avem un *game tree* în care fiecare frunză se găseşte la aceeaşi distanţă $2h$ faţă de rădăcină şi conţine o valoare booleană, iar fiecare nod intern (inclusiv rădăcina) are exact trei descendenţi direcţi (copii). Fiecare nod intern aflat la o distanţă pară faţă de rădăcină este etichetat cu AND iar fiecare nod intern aflat la o distanţă impară faţă de rădăcină este etichetat cu OR. Implementaţi în R un algoritm pentru evaluarea acestui tip de arbore (determinarea valorii booleene din rădăcină).

Indicaţie: cursul 10.

C2. (**6 puncte - Las Vegas**) Problema mariajului stabil încearcă determinarea unei aşa numite stabilităţi între două mulţimi disjuncte de acelaşi cardinal: $\mathcal{M} = \{M_1, M_2, \ldots, M_n\}$ (bărbaţi) şi $\mathcal{W} = \{W_1, W_2, \ldots, W_n\}$ (femei); un *cuplaj* este o bijecţie $f : \mathcal{M} \to \mathcal{W}$. Fiecare persoană are o lista de preferinţe a membrilor celeilalte mulţimi (listele sunt ordonate în ordinea inversă a atractivităţii). Un cuplaj este f *instabil* dacă există $M_i, M_j \in \mathcal{M}$ astfel ıncât 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*.

Implementaţi următorul algoritm aleator pentru determinarea unui cuplaj stabil.

for $(j = \overline{1,n})$ **do**
  $f(M_i) = -1$; $f^{-1}(W_i) = -1$; // **iniţial 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 şi uniform un $j \in \{1, 2, \ldots, n\}$ iarM_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 $v = v_1 v_2 \ldots v_n \in \{0, 1\}^n$, definim

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

Să presupunem cădorim săverificăm dacă un cuvânt dat $u \in \{0, 1\}^n$ aparţine unei mulţimi de cuvinte $U = \{u^1, u^2, \ldots, u^m\}$. Implementaţi următorul algoritm aleator pentru rezolvarea acestei probleme.

uniformly choose a prime number $p \leqslant n^2$; // **folosiţi funcţia** *Primes* **din pachetul** *numbers*.
$r = number(u)\ (mod\ p)$;
for $(j = \overline{1,m})$ **do**
  $r_j = number(u^j)\ (mod\ p)$;
end for
if $(r \in \{r_1, r_2, \ldots, r_m\})$ **then**
  return "$u \in U$";
end if
return "$u \notin U$";

Rezolvările acestor exerciţii (funcţiile R şi apelurile lor) vor fi redactate într-un singur script R.