cwork3b.tex 23/10/2003

MAS/202 Algorithmic Mathematics: Coursework 3

Franco Vivaldi

DEADLINE: Wednesday of week 5, at 12:00 pm.

CONTENT: Arithmetic; partitions

MicroESSAY : Write an essay on equivalence relations. [$\not\in$, 100]

Problem 1. Define concisely [$\not\in$]

(a) $\left\{\frac{a}{b} \,:\, a,b,\in\mathbb{Z},\ b\neq 0,\ b\not|a\right\}$

(b) $\{x\in\mathbb{Z} \,:\, x\,\mathrm{MOD}\,2 = 1,\ 7|x\}$

(c) $(a_1, a_2, \dots)$ $\quad a_i = p_{i+1} - p_i$, p_i the ith prime, $i\geq 1$.

Problem 2. An integer is *square-free* if it is not divisible by any square greater than 1.

(a) Find all square-free integers in the interval [40, 60].

(b) Consider the following algorithm

`Algorithm SquareFree`
`INPUT`: n, a positive integer.
`OUTPUT`: TRUE if n is square-free, FALSE otherwise.

- Explain how to use the algorithm `IntegerFactorization` to construct `SquareFree`. [$\not\in$, 50]
- Write `SquareFree`.

[*Hint:* the output of `IntegerFactorization` is a finite sequence P; denote its cardinality by $\#P$, and its elements by P_i, $i = 1,\dots,\#P$. Treat the case $n = 1$ separately.]

Problem 3. Let $X = \{1, 2, 3\}$. Determine relations R_1, R_2, R_3 on X, such that

(a) R_1 is reflexive and transitive, but not symmetric;

(b) R_2 is reflexive and symmetric, but not transitive;

(c) R_3 is symmetric and transitive, but not reflexive.

In each case, try to make relations have as few elements as possible.

Problem 4. Determine the possible cardinalities of an equivalence relation on a set of 4 elements.
[*Hint:* Analyze the possible partitions of a set with 4 elements, and for each count the number of elements of the corresponding equivalence relation.]

Problem 5. Let $\mathcal{P}$ be a partition of a finite set X, and let $P(x)$ be the part of $\mathcal{P}$ containing x.

(a) Explain why the formula $\mathcal{P} = \{P(x) : x \in X\}$ does not translate into an efficient algorithm for constructing $\mathcal{P}$ from the knowledge of X and P.
[*Hint:* Measure efficiency by the number of evaluations of the function P. How many evaluations are involved in the above formula? How could such number be reduced? Look at concrete examples involving small sets X.]

(b) Write an algorithm to the following specifications

`Algorithm Partition`

`INPUT`: X, P

`OUTPUT`: $\mathcal{P}$

You may use set operators (union, etc.), and represent the elements of a set using subscripts $A = \{A_1, A_2, \ldots\}$. Be careful that $\mathcal{P}$ is a *set of sets.*