

Milan Geryk

Algoritmy. 25. KOMBI. Algoritmus pro vytvoření kombinací f -té třídy z n prvků

Aplikace matematiky, Vol. 17 (1972), No. 1, 73--74

Persistent URL: <http://dml.cz/dmlcz/103395>

Terms of use:

© Institute of Mathematics AS CR, 1972

Institute of Mathematics of the Academy of Sciences of the Czech Republic provides access to digitized documents strictly for personal use. Each copy of any part of this document must contain these *Terms of use*.



This paper has been digitized, optimized for electronic delivery and stamped with digital signature within the project *DML-CZ: The Czech Digital Mathematics Library* <http://project.dml.cz>

ALGORITMY

25. KOMBI

ALGORITMUS PRO VYTVOŘENÍ KOMBINACÍ k -TÉ TRŽIDY Z n PRVKŮ

MILAN GERYK, Přerov, Výzkumný ústav Přerovských strojíren

Procedura vytváří postupně všechny kombinace (bez opakování) k -té třídy z n přirozených čísel $1, 2, 3, \dots, n$ tak, že do prvních k slov pole A umístí při jednom vstupu do procedury jednu vzestupně uspořádanou k -tici z těchto kombinací. Boolskou veličinu B je třeba před prvním vstupem do procedury nastavit na **true**. Přes návěští L nastává vedlejší výstup z procedury v těchto případech: 1) všechny kombinace v počtu $\binom{n}{k}$ byly již vyčerpány 2) bylo zadáno $n < k$ 3) bylo zadáno $k \leq 0$. Po výstupu návěštím L je B automaticky nastaveno opět na **true**. Mezi prvním a posledním vstupem téhož příkladu musí uživatel zajistit nedotknutelnost pole A a veličin n, k, B . Před použitím procedury *KOMBI* je nutno deklarovat pole $A [1 : M]$ typu *integer*, kde $M \geq k$.

```

procedure KOMBI ( $n, k, A, B, L$ ) ;
value  $n, k$ ; integer  $n, k$ ; integer array  $A$ ; Boolean  $B$ ; label  $L$ ;
begin integer  $I, J$ ;
  if  $B$  then
    begin if  $n < k \vee k \leq 0$  then goto  $L$ ;
      for  $I := 1$  step 1 until  $k$  do  $A[I] := I; B := \neg B$ 
    end else
      begin for  $J := k, J - 1$  while  $J \geq 1$  do
        begin  $I := k - J$ ;
          if  $A[J] - n + I \neq 0$  then
            begin  $A[J] := A[J] + 1$ ;
              for  $J := J + 1$  while  $J - 1 \neq k$  do
                 $A[J] := A[J - 1] + 1$ ; goto  $L1$ 
              end
            end ;  $B := \neg B$ ; goto  $L$ 
          end ;
        end
      L1: end

```

Příklad použití pro $n = 4, k = 2$:

$B := \text{true};$

$U: \text{KOMBI}(4, 2, A, B, V);$

$\wedge \wedge \wedge$

$\wedge \wedge \wedge$

$\wedge \wedge \wedge$

goto $U;$

$V: \wedge \wedge \wedge$

Procedura poskytne v prvých dvou slovech pole A při prvním vstupu čísla $\{1; 2\}$, při dalších vstupech postupně $\{1; 3\}, \{1; 4\}, \{2; 3\}, \{2; 4\}, \{3; 4\}$. Po následujícím vstupu do procedury nastane ihned výstup na návěští V .

Poznámka. Proceduru lze jednoduše upravit pro případ, že kombinace mají být tvořeny z n prvků pole $P[1 : n]$. Pak pro předání výsledků je potřebí navíc pole $Q[1 : M]$ stejného typu jako P . V proceduře se doplní parametry P, Q a jejich specifikace a poslední řádek změní se na tvar

L1: for $I := 1$ **step** 1 **until** k **do** $Q[I] := P[A[I]]$ **end**