Variace a kombinace s omezeným opakováním

STANISLAV TRÁVNÍČEK

Přírodovědecká fakulta UP, Olomouc

Ve stále živé publikaci [1] rozebírají autoři problematiku pojetí a náplně vyučování matematice jak z hlediska vývoje vědy, tak z hlediska tradic. Připomínají i typicky školské hledisko – *vyučovat to, co se dá lehce nacvičit a vyzkoušet.* Než se této myšlence nedůvěřivě usmějeme, nahlédněme do učiva kombinatoriky.

První učivo a první vzorec nám říká, že počet $V(k,n)$ k-členných variací z n prvků je

$$V(k,n) = n(n-1)(n-2)\dots(n-k+1). \tag{1}$$

V navazujících úlohách se tvoří vlajky z různobarevných pruhů a sestavují se čísla z různých číslic. Na variace navazují logicky permutace se vzorcem

$$P(n) = n! \tag{2}$$

V úlohách pak přehazujeme pořadí např. žáků nebo cvičenců, a tvoříme i skupiny různých číslic.

Pokračuje se učivem a vzorcem pro počet $K(k,n)$ všech k-členných kombinací z n prvků

$$K(k,n) = \binom{n}{k}. \tag{3}$$

V úlohách pak z celku vytváříme skupiny, u nichž nezáleží na pořadí prvků.

Nyní přicházejí případy, kdy některé prvky mohou být stejné, resp. že se mohou ve skupinách opakovat. Nejprve opět *variace s libovolným opakováním*, tj. vytváření uspořádaných k-členných skupin z n prvků, přičemž každý prvek se ve skupině může vyskytovat až k-krát (že nejvýše k-krát, jak říká učebnice, je samozřejmé, když jde o k-členné skupiny). Počet těchto variací nám dává hezký vzorec

$$V'(k,n) = n^k, \tag{4}$$

z něhož pro $k = n$ dostáváme vzorec pro permutace s libovolným opakováním. V úlohách se nezapomíná na důležitý případ, kdy $n = 2$ (např. nuly a jedničky).

Pokračuje se *permutacemi s omezeným opakováním*, v nichž se jednotlivé prvky mohou opakovat s různou zadanou četností k_1, k_2, …, k_n:

$$P'(k_1, k_2, \ldots, k_n) = \frac{(k_1 + k_2 + \cdots + k_n)!}{k_1!k_2!\ldots k_n!} \tag{5}$$

Oblíbenými úlohami jsou permutace písmen daného slova. Posledním tématem jsou k-členné *kombinace s libovolným opakováním* z n prvků, přičemž každý prvek se ve skupině může vyskytovat až k-krát. Jejich počet je

$$K'(k, n) = \binom{n + k - 1}{k}. \tag{6}$$

Těchto šest vzorců mají středoškolští studenti nejen znát (dnes tedy alespoň o nich vědět), ale mají i poznat, ve kterém případě mají který vzorec použít. Pokrývají tyto vzorce všechny případy, které jsou v kombinatorice (a zejména v praxi) možné?

Úloha 1

Katka má 6 lístků s písmeny; na dvou je O, na dvou P a dále má jednou E a jednou L. Kolik různých čtyřhláskových slov může ze svých písmen složit?

Řešení. Na tuto úlohu nelze použít žádný z uvedených šesti vzorců. Jde zřejmě o variace 4. třídy ($T = 4$) ze šesti prvků ($N = 6$), protože u slova záleží na pořadí písmen, přitom však různé prvky mají různou možnost opakování. Jde tedy o případ, který lze označit jako variace s omezeným opakováním.

Pro názornost zapišme řetězec:

O O P P E L

Zjistěme, kolik variací dostaneme. Vidíme, že je několik možností podle toho, co daná čtveřice obsahuje. Předně to mohou být čtveřice z písmen O O P P, kterých je podle vzorce (5)

$$P_1 = P'(2, 2) = \frac{4!}{2!2!} = 6.$$

Pak vezměme případ, kdy ve slově bude celá jedna dvojice (jsou dvě možnosti – OO nebo PP) a k tomu dvojice zbývajících písmen po jednom, podle (1) je těchto dvojic $3 \cdot 2 = 6$. Těchto případů je proto

$$P_2 = 2 \cdot P'(2,1,1) \cdot 3 \cdot 2 = 2 \cdot \frac{4!}{2!} \cdot 6 = 72.$$

Zbývá případ, že máme 4 různá písmena, těchto možností je podle (2)

$$P_3 = P(4) = 4! = 24.$$

Celkový počet slov je

$$P = P_1 + P_2 + P_3 = 6 + 72 + 24 = 102.$$

Vidíme, že na citovaném výroku z knihy [1] něco je: na případy, jako je tento z naší úlohy 1, nemáme vzoreček a tak se jimi při výuce matematiky nezabýváme; přitom okolnost, že máme různých věcí různý počet, je v praxi docela přirozená. Ukázali jsme, že se těmto případům vyhýbat nemusíme, stačí provést roztřídění různých možností a v každé té třídě použít vhodný z výše uvedených vzorců.

S určením počtu variací s omezeným opakováním i s jejich výpisem si dobře dovede poradit výpočetní technika, tedy vhodný počítačový program. Učitel matematiky, který zná základy programování, si pro svou potřebu může takový program sestavit. Je samozřejmé, že algoritmus programu nesleduje nutně naše lidské uvažování, ale ze zásoby svých znalostí zvolíme takové prostředky, které jsou pro počítač jednodušší a vhodnější; rekurze použitá v následujícím programu `VarOmOp` – je však lidskému postupu blízká (permutace zde bereme jako zvláštní případ variací, když se třída variace shoduje s počtem všech prvků). Ježto nás nezajímá jen počet variací, ale i samotné variace, zjistí náš program počet variací tak, že je prostě všechny vytvoří a vypíše „podle abecedy“ stanovené zadáním. V našem případě platí abeceda, v níž je pořadí písmen dané zadáním úlohy 2, symbolicky O < P < E < L. Uveďme začátek hledaného výpisu:

OOPP, OOPE, OOPL, OOEP, OOEL, OOLP,
OOLE, OPOP, OPOE, OPOL, OPPO, ...

Vidíme, že když písmena nejsou uspořádána podle tradiční abecedy, vyžaduje výpis podle dané dočasné abeced od člověka velkou pozornost.

Náš program načte řetězec C znaků (jehož maximální délka je 8), a tvoří třídy o délce T. Vytvoření všech variací zařizuje procedura `Vytvor`

(Vytvoř) se dvěma řetězcovými parametry A, B: do prvního se postupně skládají vytvářené variace, ze druhého se postupně čerpají potřebné znaky, tj. na počátku je řetězec A prázdný a v B je celý načtený řetězec C. V průběhu práce, pokud je do A přidán i-tý znak z B, je pro další krok rekurze (pro připojení dalšího znaku) tento znak z řetězce B vypuštěn, protože už byl použit. Když má řetězec A délku T, je daná variace vytvořena a tisková procedura ji vytiskne. Ta rovněž zařídí, že se běh programu po každých třiceti variacích zastaví, aby si uživatel mohl vytvořené variace prohlédnout (každá variace je opatřena pořadovým číslem) a pro přehlednost je na obrazovce využito barevného odlišení různých fází práce.

```
program VarOmOp;
uses Crt;

const
  MaxN = 8;
  BNad = yellow+16*red;
  BInf = lightgreen;
  BVst = white;
  BVys = yellow;
var
  C: string;
  K, T, N, Radek: Integer;

procedure Tiskni(Q: string);
var I: Integer;
begin
  TextAttr := BVys;
  Inc(Radek);
  for I := 1 to T do Write(' ',Q[I]);
  TextAttr := BInf;
  Write('   ', Radek);
  if (Radek mod 30 = 0) then
  begin
    GotoXY(2,20);
    Write('Enter');
    ReadLn
  end;
  GotoXY(1+3*(T+1)*((Radek mod 30) div 10),9+(Radek mod 10));
end; {Tiskni}

procedure Vytvor(A, B: string);
var
  J: Integer;
begin
  if Length(A) >= T then Tiskni(A)
```

```
  else
    for J := 1 to Length(B) do
      if Pos(B[J], B) = J then
      Vytvor(A + B[J], Copy(B,1,J-1)+Copy(B,J+1, N));
end;    {Vytvor}

begin  {program}
  ClrScr;
  WriteLn; TextAttr := BNad;
  WriteLn('Variace s omezenym opakovanim znaku retezce');
  WriteLn; TextAttr := BInf;
  WriteLn('Kazdy znak zadej tolikrat, kolikrat se muze opakovat');
  WriteLn; TextAttr := BVst;
  Write('Vstupni retezec (nejvyse ',MaxN,') znaku:');
  ReadLn(C);
  N := Length(C);
  if N > MaxN then N := 2;
  Write(' Trida variace: ');
  ReadLn(T);
  if T > N then T := 1;
  WriteLn; TextAttr := BVys;
  Radek := 0;
  GotoXY(1,9);
  Vytvor('', C);
  for K := 1 to T+2 do Write(' *');
  GotoXY(2,20);
  TextAttr := BInf;
  WriteLn('Konec.');
  ReadLn
end.  {progran}
```

Pokud při spuštění programu volíme T rovno délce řetězce, dostaneme výpis všech permutací.

Nyní se zabývejme kombinacemi s omezeným opakováním.

Úloha 2

V cyklistické stáji jsou 3 Francouzi, 2 Španělé, 2 Holanďané, 1 Belgičan a 1 Čech. Kolika způsoby, pokud jde jen o národnosti, lze pro příští závod sestavit pětičlenný tým?

Řešení.

Má se tedy stanovit počet kombinací 5. třídy z pěti různých prvků s omezeným opakováním (tj. prvků je sice 9, ale některé jsou navzájem shodné). Stejně jako při řešení úlohy 1 lze i zde získat řešení roztříděním na různé možnosti. Označme národy F, S, H, B, C a dále X, Y, Z nechť jsou navzájem různé národy. Pak můžeme postupně v jednotlivých případech skladby týmu zjišťovat počet možností, například takto:

a) 3 F + 2 X (X je S nebo H) 2 (FFFSS, FFFHH)

b) 3F + X + Y 6 (X, Y – dva národy ze čtyř)

c) 2X + 2Y + Z $3 \times 3 = 9$

(třemi způsoby se vytvoří dvojice, třemi způsoby se tým doplní)

d) 2X + Y + Z + U $3 \times 4 = 12$

(mohou být tři různé dvojice, a ze zbývajících čtyř národů se vždy jeden vynechá)

e) X + Y + Z + U + V 1 (závodníci různých národů)

Celkem je 30 možností.

Program `KomOmOp` pro výpočet počtu kombinací s omezeným opakováním a výpis všech takových kombinací se od předchozího `VarOmOp` samozřejmě odlišuje. Například místo dvanácti variací OOPE, OOEP, OPOE, OPEO, OEOP, OEPO, POOE, POEO, PEOO, EOOP, EOPO, EPOO dostaneme jedinou permutaci. Je proto třeba, aby se již jednou vytvořená a vytištěná permutace nedostala do výstupu znovu.

Zvolili jsme zde jiný postup, kdy tvoříme N-tice nul a jedniček, v nichž je právě T jedniček (viz pole B), což kontroluje procedura `Pocet1`. K jejich vytváření je použita rekurzivní procedura `Vytvor01`. Tisková procedura navíc kontroluje, aby daná kombinace byla započítána a vytištěna jen jednou. Úprava obrazovky je podobná jako u předchozího programu s tím, že výsledky jsou ve dvou sloupcích po 15 (ježto jsme zvolili jsme větší `MaxN = 10`).

```
program KomOmOp;
uses Crt;

const
  MaxN = 10;
  BNad = yellow+16*red;
  BInf = lightgreen;
  BVst = white;
  BVys = yellow;

var
  T, N, P, R, Radek: Integer;
  C: string;
  A: array[1..MaxN] of Integer;
```

```
procedure Tiskni;
var Q: Integer;
begin
  TextAttr := BVys;
  Inc(Radek);
  for Q := 1 to N do
    if A[Q] = 1 then Write(' ',C[Q]);
  TextAttr := BInf;
  Write('  ',Radek);
  if Radek mod 30 = 0 then
  begin
    GotoXY(2,20);
    Write('Enter');
    ReadLn
  end;
  GotoXY(1+3*(T+1)*((Radek mod 30) div 15),9+(Radek mod 15));
end;

procedure Rozhodni;
var
  J, L: Integer;
  Tisk: Boolean;
begin
  Tisk := true;
  J := 1;
  repeat
    if A[J] = 1 then
    begin
      for L := 1 to J do
        if (C[L] = C[J]) and (A[L] = 0) then
          Tisk := false
    end;
    Inc(J)
  until (J > N) or not Tisk;
  if Tisk then Tiskni
end; {Rozhodni}

procedure Pracuj(K, P: Integer);
var I, S: Integer;
begin
  if K <= N then
  begin
    if P <= T then
    begin
      A[K] := 1;
      Pracuj(K+1,P+1)
    end;
```

```
      A[K] := 0;
      Pracuj(K+1,P)
    end
    else
      if P = T then Rozhodni
  end;  {Pracuj}

begin {program}
  ClrScr; WriteLn;
  TextAttr := BNad;
  WriteLn(' Kombinace s opakovanim znaku retezce ');
  WriteLn; TextAttr := BInf;
  WriteLn(' Kazdy znak zadej tolikrat,
    kolikrat se muze opakovat.');
  WriteLn; TextAttr := BVst;
  Write(' Vstupni retezec ');
  TextAttr := BInf;
  Write('(nejvyse ',MaxN,' znaku): ');
  TextAttr := BVst;
  ReadLn(C);
  N := Length(C);
  if N > MaxN then N :=2;
  Write(' Trida kombinace: ');
  ReadLn(T);
  if not (T in [1..N]) then T := 1;
  WriteLn; TextAttr := BVys;
  Radek := 0;
  GotoXY(1,91);
  Pracuj(1,0);
  for R := 1 to 2*T+5 do Write('*');
  GotoXY(2,20);
  TextAttr := BInf;
  WriteLn('Konec.');
  ReadLn
end. {program}
```

Může se zdát, že zbývá ještě jeden neřešený případ, a to, že některých prvků máme libovolný počet a jiných jen omezený počet. Ten však lehce převedeme na předchozí, stačí, když prvky, kterých máme libovolná počet, zapíšeme do řetězce v počtu T. Pokud učitel matematiky při probírání kombinatoriky přijme do výuky (aspoň jako zajímavost) i případy omezeného opakování, může zde uvedené programy použít ke kontrole výsledků.

Literatura

[1] *Fischer, R. – Malle, G.*: Člověk a matematika. SPN, Bratislava, 1992.