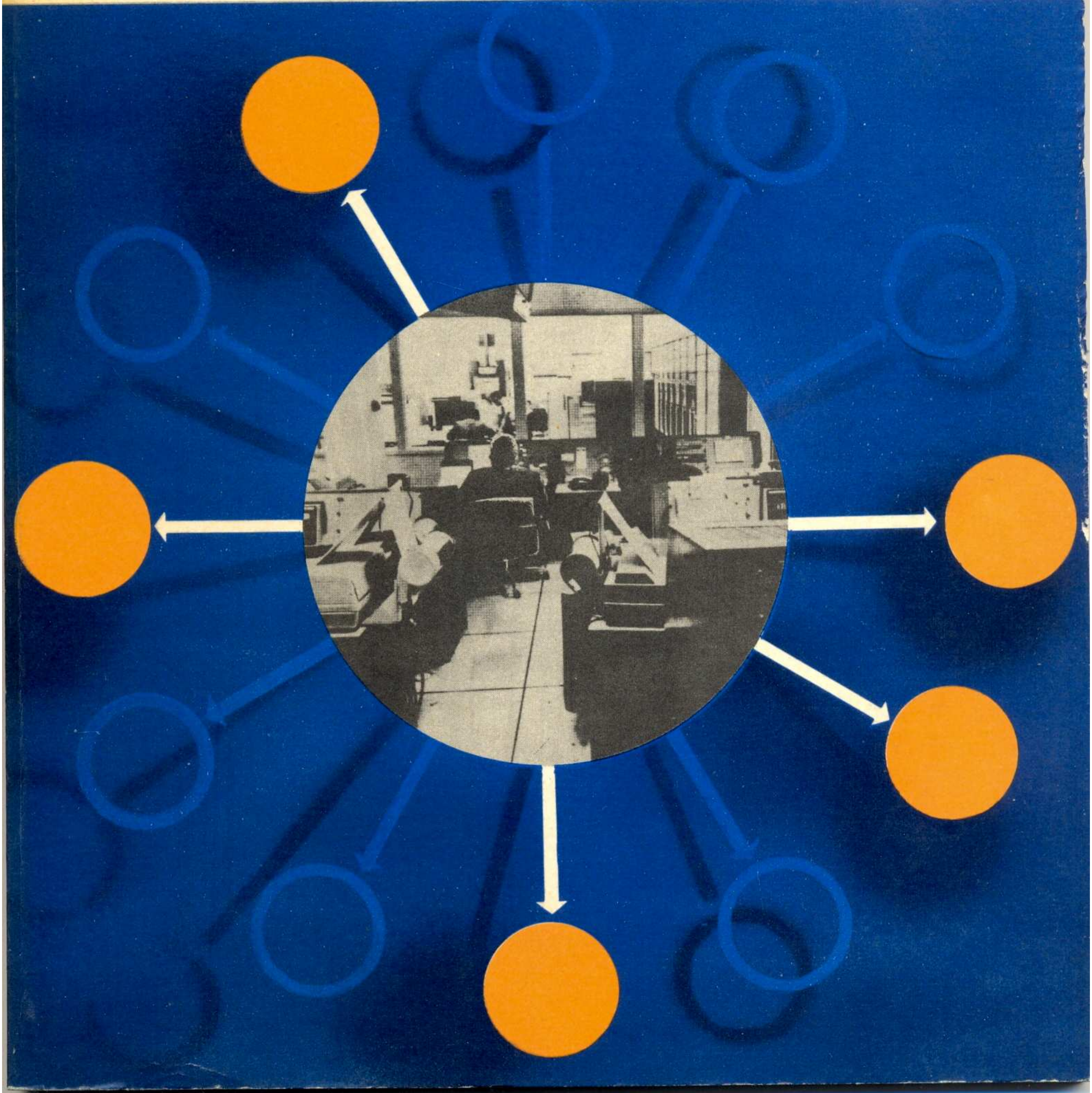
INFORMÁCIÓ ELEKTRONIKA

1975/1



Algoritmusok az összes n-edfokú permutáció előállítására

I. rész

Dénes Tamás

Számos matematikai és gyakorlati problémában fordul elő n elem különböző permutációinak a felhasználása. Példaként megemlíthetők a közgazdasági számításoknál előforduló mátrix műveletek, a távközlésben jelentkező kódolási feladatok.
Az n-edfokú permutációk előállítására számos módszert dolgoztak ki, lásd pl. [1] ,[2], [4]. Tekintettel arra, hogy már elég kis n esetén is rendkívül nagy a permutációk száma, az előállítást célszerű számítógéppel végezni. Igy előtérbe terülnek a permutációkat előállító algoritmusokkal szemben támasztott követelmények, a nagy előállítási sebesség és a kis memóriaigény.
Jelen dolgozat első részében az összes n-edfokú permutáció előállítására mutatunk be egy rekurzív eljárást, amely előnyös a számítógépes realizáció szempontjából.
A dolgozat második része egy kölcsönösen egyértelmű megfeleltetést ad az összes n-edfokú permutáció és $n!$ darab természetes szám között, amelyből a permutációk előállítására egy direkt eljárás nyerhető, valamint megadjuk a rekurziós módszer számítógépes programjának blokkdiagramját, és az IBM 360/40 típusú számítógépen végzett futási eredményeket.

Alapfogalmak, jelölések

Kölcsönösen egyértelmű leképezés: Olyan leképezés, amelynél minden elem fellép képként és különböző elemeknek különböző a képük.
Leképezések szorzása: Két vagy több leképezés egymás-utáni végrehajtása.
Permutáció: Egy n elemű halmaz önmagára történő leképezése (n elem valamilyen sorrendben történő felírása).
Transzpozíció: Olyan ciklus, amelynek a hossza kettő.
Elemi transzpozíció: $(i\ i{+}1)$ alakú transzpozíció
Ciklus (ciklikus permutálás): Olyan permutálás, amelynél a permutálandó elemeket alkalmas sorrendben felírva, mindegyik helyébe a következő, s az utolsó helyébe az első kerül.
Ciklus hossza: A ciklusban levő elemek száma.
Idegen ciklusok: Olyan ciklusok, amelyeknek nincs közös elemük.
Identikus permutálás: Olyan permutáció, amelyben minden elem képe önmaga.

Jelölések

Permutáció: $\pi, \begin{pmatrix} i_1 i_2 \dots i_n \\ j_1 j_2 \dots j_n \end{pmatrix}, a_1 a_2 \dots a_n, I$

Transzpozíció: τ, (ij)

Rekurziós eljárás

Az eljárás elemi transzpozíciók segítségével állítja elő a permutációkat. A rekurziós formula a következő:

$$\pi_1 = \begin{pmatrix} 12\dots n \\ 12\dots n \end{pmatrix} \qquad (1)$$

A későbbiek során látható, hogy π_1 tetszőleges rögzített permutáció lehet.

$$\pi_{m+1} = \sigma_m \cdot \pi_m \qquad (2)$$

A (2) képlet alkalmazásához tehát σ_m meghatározása szükséges. (n-edfokú permutációk esetén $1 \leqslant m \leqslant n!$)
Osztályozzuk a természetes számokat az $A_1, A_2, \dots, A_k$ osztályokba úgy, hogy

$$m \in A_k \Leftrightarrow k! \mid m \text{ és } (k+1)! \nmid m \qquad (3)$$

σ_m-re következő definiciót adjuk:

$$\sigma_m = \begin{cases} \tau_1 \cdot \tau_3 \cdot \tau_5 \cdot \dots \cdot \tau_k \text{ ha } m \in A_k \text{ és } k \text{ páratlan} \\ \tau_2 \cdot \tau_4 \cdot \tau_6 \cdot \dots \cdot \tau_k \text{ ha } m \in A_k \text{ és } k \text{ páros} \end{cases} \qquad (4)$$

A (4) definícióban $\tau_1, \tau_2, \dots \tau_k$ elemi transzpozíciókat jelölnek, ahol $\tau_i = (i\ i+1)$. Ily módon tehát σ_m egy transzpozicíószorzatot jelöl, melyet ezután σ_m transzformációnak nevezünk, amely szorzat tényezőinek a számát az szabja meg, hogy m a (3) definíció szerint melyik osztályba tartozik.
Másképpen (a $\tau_i = (i\ i{+}1)$ jelölést használva) (4) így írható:

$$\sigma_m = \begin{cases} (12)(34)(56) \cdot \dots \cdot (k\ k+1) \\ \qquad \text{ha } m \in A_k \text{ és } k \text{ páratlan} \\ (23)(45)(67) \cdot \dots \cdot (k\ k+1) \\ \qquad \text{ha } m \in A_k \text{ és } k \text{ páros.} \end{cases} \qquad (5)$$

Példaként nézzük meg, hogy hogyan áll elő az összes negyedfokú permutáció *(1. táblázat)*.

1. táblázat

$m+1$	k	σ_m	π_{m+1}	π_m+1 (Az őselemek nélkül)
1	0		1234 1234	1234
2	1	(12)	1234 2134	2134
3	2	(23)	1234 2314	2314
4	1	(12)	1234 3214	3214
5	2	(23)	1234 3124	3124
6	1	(12)	1234 1324	1324
7	3	(12) (34)	1234 3142	3142
8	1	(12)	1234 1342	1342
9	2	(23)	1234 1432	1432
10	1	(12)	1234 4132	4132
11	2	(23)	1234 4312	4312
12	1	(12)	1234 3412	3412
13	3	(12) (34)	1234 4321	4321
14	1	(12)	1234 3421	3421
15	2	(23)	1234 3241	3241
16	1	(12)	1234 2341	2341
17	2	(23)	1234 2431	2431
18	1	(12)	1234 4231	4231
19	3	(12) (34)	1234 2413	2413
20	1	(12)	1234 4213	4213
21	2	(23)	1234 4123	4123
22	1	(12)	1234 1423	1423
23	2	(23)	1234 1243	1243
24	1	(12)	1234 2143	2143

A következőkben rátérünk az algoritmus részletes ismertetésére és annak bizonyítására, hogy ezzel az eljárással valóban előállítható $n!$ darab n-edfokú permutáció, és ezek mind különbözőek lesznek.

A.1.

Az algoritmus a következő transzformációszorzatot állítja elő (1) és (2) szerint:

$$\pi_{k+1} = \sigma_k \cdot \sigma_{k-1} \cdot \ldots \cdot \sigma_2 \cdot \sigma_1 \cdot \pi_1 \quad k = 1, \ldots, n!-1 \tag{6}$$

Osztályozzuk a σ_m transzformációkat az indexük szerint, vagyis két transzformáció σ_r és σ_s akkor és csak akkor tartozik egy osztályba, ha r és s a (3) definíció szerint egy osztályba tartozik.

Ekkor könnyen belátható, hogy az egy osztályba tartozó transzformációk egyenlők, hiszen r és s egyértelműen határozzák meg a (4) definíció k osztályindexét, vagyis a σ_s illetve σ_r-ben szereplő transzpozíciókat.

A.2.

A (3)-ban és A.1.-ben definiált osztályozás és a transzformációk egyenlősége miatt igaz a következő:

$$\begin{array}{llll}
\sigma_1 & = \sigma_{(n-1)!+1} = \sigma_{2[(n-1)!]+1} = \ldots & = \sigma_{(n-1)[(n-1)!]+1} \\
\sigma_2 & = \sigma_{(n-1)!+2} = \sigma_{2[(n-1)!]+2} = \ldots & = \sigma_{(n-1)[(n-1)!]+2} \\
\cdot & \cdot \quad \cdot & \cdot \\
\cdot & \cdot \quad \cdot & \cdot \\
\cdot & \cdot \quad \cdot & \cdot \\
\sigma_{(n-1)!} & = \sigma_{2[(n-1)!]} = \sigma_{3[(n-1)!]} = \ldots & = \sigma_{n[(n-1)!]}
\end{array} \tag{7}$$

A (7) egyenlőségrendszer $n = 4$ esetén figyelemmel kísérhető az 1. táblázat 3. (σ_m) oszlopában.

A.3.

A (6) és (7) egyenlőségeket összevetve azt kapjuk, hogy az algoritmus $(n-1)!$ szerint ciklikusságot mutat. Helyettesítsük ugyanis be a (7) egyenlőségrendszerben kapottakat (6)-ba.

$$\pi_{n!} = \sigma_{(n-1)!-1} \cdot \ldots \cdot \sigma_1 \cdot [\sigma_{(n-1)!} \cdot \ldots \cdot \sigma_1]^{n-1} \cdot \pi_1 \tag{8}$$

Mivel a $\sigma_{(n-1)!-1} \cdot \ldots \cdot \sigma_1$ transzformációszorzat nem más, mint az összes $(n-1)$-edfokú permutáció előállításához szükséges transzformációk szorzata, így az egyszerűbb jelölés miatt vezessük be a következőt:

$$\rho_r = \sigma_{r!-1} \cdot \ldots \cdot \sigma_1 \tag{9}$$

Ahol tehát ρ_r az összes r-edfokú permutáció előállításához szükséges transzformációk szorzatát jelöli. Ekkor tehát (8) így írható:

$$\rho_n = \rho_{n-1} \cdot [\sigma_{(n-1)!} \cdot \rho_{n-1}]^{n-1} \cdot \pi_1 \tag{10}$$

A.4.

Az eddigiekből kiderül, hogy az eljárás egy $n!-1$ darab transzformációból álló szorzatot állít elő, amelyből a szorzás értelmezése szerint a permutációk úgy állnak elő, hogy jobbról balra haladva tényezőnként végrehajtjuk a transzformációkat és minden lépés egy új permutációt ad (lásd (2)), így a π_1 kezdőpermutációt beleszámítva valóban $n!$ darab permutációt kapunk. A (10) formulából látható, hogy az eljárás egymásba skatulyázott transzformációszorzatokból áll, melyek ismételt végrehajtását írja elő. Mivel tényezőnként véve előállítja az összes r-edfokú permutációt, ábrázoljuk ρ_r-t egy téglalappal és nevezzük r-edfokú blokknak. Így a (10) formula jól szemléltethető az *1. ábra* szerint, ahol az azonos nagyságú téglalapok azonos fokszámú blokkot jelölnek.

A *2. ábra* $n = 4$ esetén mutatja a blokkos elrendezést, amelyből látható, hogy a blokkok utolsó oszlopában azonos elemek állnak. Ezt a III. részben fogjuk kihasználni. Most tekintsük a (10) formula szögletes zárójelében álló kifejezést.

σ_{n-1} értelmezése szerint $\sigma_{(n-1)!}\ \rho_{n-1}$ a második n-edfokú blokk első permutációját előállító transzformációszorzat.

Ahhoz, hogy valóban $n!$ darab különböző permutációt állítson elő ρ_n, szükséges, hogy $\sigma_{(n-1)!}\ \rho_{n-1}$ különböző hatványai különböző permutációt állítsanak elő. (Ez egyben elegendő feltétel is.)

A.5.

Állítás

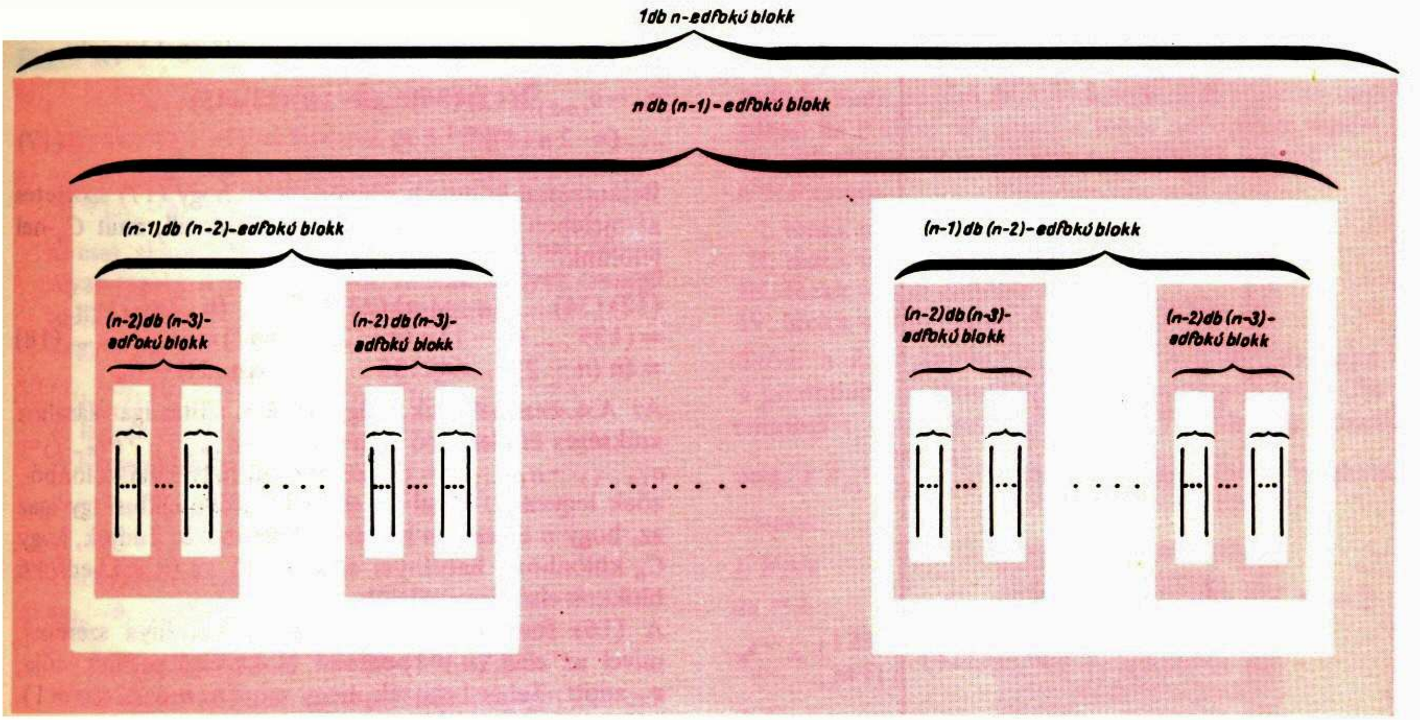
Az (1), (2) illetve (10) formulával leírt algoritmus $n!$ darab különböző permutációt állít elő.

Bizonyítás

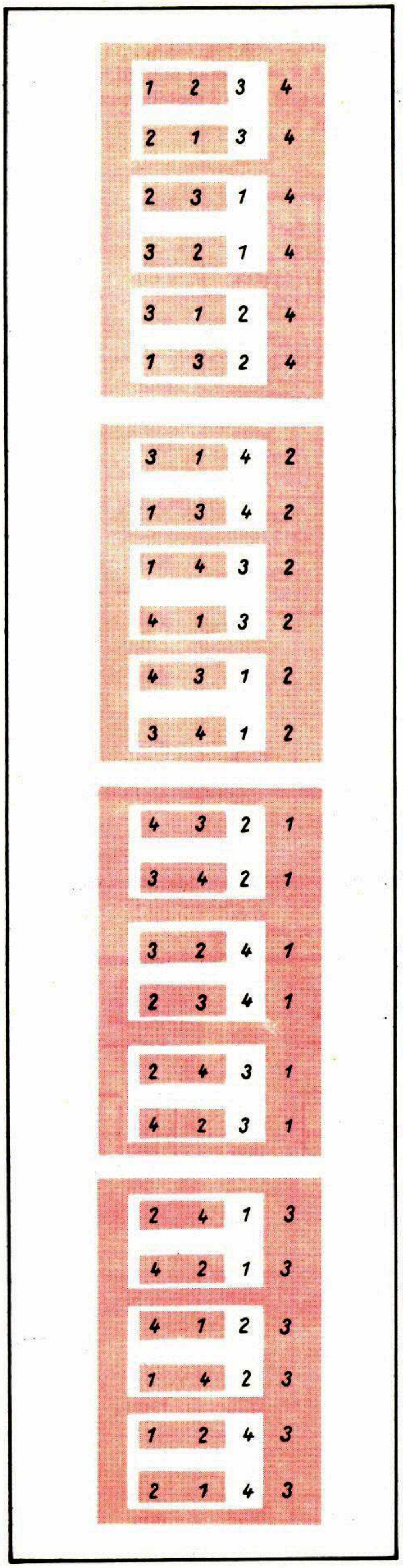
A bizonyítást teljes indukcióval végezzük.

Az állítás $k = 2$, $k = 3$, $k = 4$ esetén teljesül; ezt a 2. ábrán figyelemmel kísérhetjük. $k = 3$ esetében az alábbi ábra mutatja az előállítást.

$m+1$	k	σ_m	π_{m+1}
1	0	–	$\begin{pmatrix}123\\123\end{pmatrix}$
2	1	(12)	$\begin{pmatrix}123\\213\end{pmatrix}$
3	2	(23)	$\begin{pmatrix}123\\231\end{pmatrix}$
4	1	(12)	$\begin{pmatrix}123\\321\end{pmatrix}$
5	2	(23)	$\begin{pmatrix}123\\312\end{pmatrix}$
6	1	(12)	$\begin{pmatrix}123\\132\end{pmatrix}$



1. ábra



2. ábra

Tegyük fel, hogy $k \leqslant (n-1)$-re igaz az állítás. Valamint tegyük fel, hogy n páros. (Páratlan n-re hasonló a bizonyítás).
Ekkor

$$\rho_{n-1} = \rho_{n-2} \cdot [\sigma_{(n-2)!} \cdot \rho_{n-2}]^{n-2} \cdot \pi_1 \tag{11}$$

formulával $(n-1)!$ különböző permutációt állíthatunk elő. Szorozzuk be balról a (11) egyenlőség mindkét oldalát $\sigma_{(n-2)!}$-al.

$$\sigma_{(n-2)!} \cdot \rho_{n-1} = [\sigma_{(n-2)!} \cdot \rho_{n-2}]^{n-1} \cdot \pi_1 \tag{12}$$

A (12) egyenlet jobb oldalán az indukciós feltevés miatt az identikus permutáció (I) áll.

$$\sigma_{(n-2)} \cdot \rho_{n-1} = \mathrm{I} \tag{13}$$

Igy (13)-ból ρ_{n-1}-re a következőt kapjuk, ha $\sigma_{(n-2)!}$ inverzével balról szorozzuk az egyenlőséget:

$$\rho_{n-1} = [\sigma_{(n-2)!}]^{-1} \tag{14}$$

$[\sigma_{(n-2)!}]^{-1}$ az (5) definició szerint nem más, mint $[(23)\ (45)\ (67)\ \ldots\ (n-2\ n-1)]^{-1}$, mivel azonban elemidegen transzpozíciók szorzatáról van szó, ennek inverze megegyezik az eredeti szorzattal, vagyis

$$\rho_{n-1} = [\sigma_{(n-2)!}]^{-1} = \sigma_{(n-2)!} = (23)(45)(67)\ldots$$
$$\ldots(n-2\ n-1) \tag{15}$$

Helyettesítsük tehát a kapott (15) eredményt (10)-be.

$$\rho_n = \sigma_{(n-2)!}[\sigma_{(n-1)!} \cdot \sigma_{(n-2)!}]^{n-1} \cdot \pi_1 \tag{16}$$

Ha figyelembe vesszük, hogy $\sigma_{(n-1)!} = (12)\,(34)\,(56)\ldots$ $\ldots(n-1 n)$, akkor a következő egyenlőséget kapjuk:

$$\rho_n = \sigma_{(n-2)!}[(12)\,(34)\ldots(n-1\,n)(23)(45)\ldots$$
$$\ldots(n-2\,n-1)]^{n-1} \cdot \pi_1 \tag{17}$$

Beszorzással könnyen ellenőrizhető, hogy (17) szögletes zárójelében egy teljes n-edfokú ciklus áll, amit C_n-nel jelölünk.

$$\begin{aligned} &(12)\,(34)\ldots(n-1\,n)\,(23)\,(45)\ldots(n-2n-1)= \\ &=(135\ldots(n-1)\,n\,(n-2)\ldots 642)= \\ &=(n\,(n-2)\ldots 642135\ldots(n-3)\,(n-1))=C_n \end{aligned} \tag{18}$$

Az A.4.-ben beláttuk, hogy az A.5. állítás igazolásához szükséges és elegendő feltétel, hogy $\sigma_{(n-1)!} \cdot \rho_{(n-1)} = \sigma_{(n-1)!} \cdot \sigma_{(n-2)!} = C_n$ különböző hatványai különbözőek legyenek. Mivel C_n teljes n-edfokú ciklus, így igaz az, hogy n és csak n hatványa különböző. Tudjuk, hogy C_n különböző hatványai állítják elő az $(n-1)$-edfokú blokkok első permutációit.
A (16) formulában C_n-nek $n-1$ hatványa szerepel, mivel az első $(n-1)$-edfokú blokk első permutációja, π_1 adott. Tehát beláttuk, hogy mind az n darab $(n-1)$-edfokú blokk első permutációja különböző lesz, ami állításunk igazságát bizonyítja.

Kölcsönösen egyértelmű megfeleltetés az összes n-edfokú permutáció és n! darab természetes szám között

Ebben a fejezetben adunk egy megfeleltetést, amely az $1, 2, \ldots n!$ természetes számok mindegyikéhez egy és csak egy n-edfokú permutációt rendel hozzá. Ez lehetővé teszi, hogy csak bizonyos tulajdonságú permutációkat állítsunk elő, valamint véletlen permutációk generálására is alkalmazható. A jelölések megegyeznek az előző fejezetben használtakkal.

B.1.

Legyenek m és b_i természetes számok, melyekre teljesül, hogy

$$1 \leqslant m \leqslant n! \tag{20}$$

$$0 \leqslant b_i \leqslant i-1 \qquad i = 2, 3, \ldots, n \tag{21}$$

Állítás

Bármely m esetén, amely (20)-nak eleget tesz, egy és csak egy b_i számsorozat létezik, melyre teljesül, hogy

$$m-1 = \sum_{i=2}^{n} b_i\,[(i-1)!] \tag{22}$$

Könnyen belátható, hogy minden (20) tulajdonságú m természetes számhoz létezik (22) tulajdonságú b_i sorozat, hiszen (22) nem más, mint m-nek faktorális számrendszerben felírt alakja.
Most bebizonyítjuk, hogy m egyértelműen meghatározza a b_i sorozatot. A bizonyítás indirekt úton történik.

Feltételek:

1. $\sum_{t=2}^{n} a_t\,[(t-1)!] = \sum_{t=2}^{n} b_t\,[(t-1)!]$
2. létezik $1 < k \leqslant n$ amelyre $a_{i_1} \neq b_{i_1}, \ldots, a_{i_k} \neq b_{i_k}$
3. $\min(i_1, i_2, \ldots, i_k) = i_1$

Ekkor az 1, 2 feltételek miatt
$(a_{i_1} - b_{i_1})(i_1-1)! + \ldots + (a_{i_k} - b_{i_k})(i_k-1)! = 0$
Az egyenletet $(i_1-1)!$-al leosztva (3 miatt)
a) $a_{i_1} - b_{i_1} + (a_{i_2} - b_{i_2})(i_2-1)(i_2-2)\cdot \ldots \cdot(i_1+1)\cdot i_1 + \ldots$
$\ldots + (a_{i_k} - b_{i_k})(i_k-1)(i_k-2)\cdot \ldots \cdot (i_1+1)\cdot i_1 = 0$

a) csak akkor lehetséges, ha az egyenlőség bal oldalán olyan tagokból álló csoportok vannak, melyek összege nulla (és bennük ilyen már nincs több).
Legyen egy ilyen
b) $(a_{j_1} - b_{j_1})(j_1-1)! + \ldots + (a_{j_p} - b_{j_p})(j_p-1)! = 0$
és legyen $\min(j_1, j_2, \ldots, j_p) = j_1$
A b) egyenlőséget leosztva $(j_1-1)!$-al:
$a_{j_1} - b_{j_1} + \ldots + (a_{j_p} - b_{j_p})(j_p-1)(j_p-2)\cdot \ldots$
$\ldots \cdot (j_1+1)\cdot j_1 = 0$
átrendezve az egyenletet:
c) $(a_{j_2} - b_{j_2})(j_2-1)(j_2-2)\cdot \ldots \cdot j_1 + \ldots$
$\ldots + (a_{j_p} - b_{j_p})(j_p-1)\cdot \ldots \cdot j_1 = b_{j_1} - a_{j_1}$
Mivel ebben már nem szerepel nulla összeget adó csoport, így az egyenlet két oldalának abszolút értékben meg kell egyeznie, de (21) miatt $|b_{j_1} - a_{j_1}| < j_1 - 1$
A c) egyenlet bal oldala viszont nagyobb j_1-nél. Tehát az állítás hamis, vagyis $a_{i_1} = b_{i_1}, \ldots, a_{i_k} = b_{i_k}$

B.2.

Az előző fejezetben ismertetett rekurziós algoritmus konstrukcióját kísérjük figyelemmel (lásd 2. ábra).

$\boxed{1}$ Minden blokk utolsó oszlopában egyenlő elemek vannak.

$\boxed{2}$ Azonos fokszámú, de különböző blokkok utolsó oszlopai különbözők. Például a 2. ábrán a harmadfokú blokkok utolsó oszlopaiban rendre a 4, 2, 1, 3 számok állnak.

Az $\boxed{1}$ és $\boxed{2}$ megállapításokból következik, hogy egy tetszőleges $a_1 a_2 \cdots a_n$ n-edfokú permutáció minden eleme meghatároz egy blokkot. Mégpedig az a_j elem egy $(j-1)$-edfokú blokkot.
Például a 3241 permutáció elemei a következő blokkokat határozzák meg:
1: III. harmadfokú blokk
4: II. másodfokú blokk
2: I. elsőfokú blokk
Látható, hogy a blokkok egymásba skatulyázása miatt, az utolsó $n-1$ elem egyértelműen meghatározza a permutációt. Jelentse tehát (22)-ben b_i a következőt.

Legyen $\pi = a_1 a_2 \cdots a_i \ldots a_n$ egy n-edfokú permutáció. Ekkor a_i egyértelműen meghatároz egy $(i-1)$-edfokú blokkot, legyen ez a k-adik (ahol $1 \leqslant k \leqslant i$).
Ebben az esetben $b_i = k - 1$.
Látható, hogy b_i-nek ez a definíciója nem mond ellent a (21) definíciónak.

B.3.

Megállapítottuk A.5.-ben, hogy a C_n ciklus hatványai állítják elő az $(n-1)$-edfokú blokkok első permutációit. Ezek viszont utolsó elemükkel egyértelműen meghatározzák a blokkok utolsó oszlopát, tehát C_n meghatározza az $(n-1)$-edfokú blokkok utolsó oszlopait, vagyis a blokkok sorrendjét.

Például $n=4$ esetén $C_4 = (4213)$. Mivel $\pi_1 = \begin{pmatrix} 1\,2 \ldots n \\ 1\,2 \ldots n \end{pmatrix}$ tudjuk, hogy az első $(n-1)$-edfokú blokk utolsó oszlopában az n szám áll, tehát a többi oszlopában rendre a C_n ciklusban n-től jobbra szereplő számok lesznek.
$n = 4$ esetén

I. blokk utolsó oszlop : 4
II. blokk utolsó oszlop : 2
III. blokk utolsó oszlop : 1
IV. blokk utolsó oszlop : 3

Tehát a B.2. pontban értelmezett k nem más, mint a π permutáció a_i elemének a C_i ciklusbeli sorszáma. (Ha a számozást balról jobbra végezzük.) Példaként nézzük meg a $\pi_m = \begin{pmatrix} 1234 \\ 2431 \end{pmatrix}$ permutáció sorszámának meghatározását.

1. lépés

$a_4 = 1 \qquad C_4 = (4213) \qquad k_1 = 3 \qquad b_4 = 2$

$\pi'_m = \begin{pmatrix} 1234 \\ 2431 \end{pmatrix} (4213)^{-2} = \begin{pmatrix} 1234 \\ 3124 \end{pmatrix}$

2. lépés

$a'_3 = 2 \qquad C_3 = (312) \qquad k_2 = 3 \qquad b_3 = 2$

$\pi''_m = \begin{pmatrix} 1234 \\ 3124 \end{pmatrix} (312)^{-2} = \begin{pmatrix} 1234 \\ 1234 \end{pmatrix}$

3. lépés

$a''_2 = 2 \qquad C_2 = (21) \qquad k_3 = 1 \qquad b_2 = 0$

Tehát $m-1 = 2\,3! + 2\,2! + 0\,1!$
$m = 17$

(Az eredmény az 1. táblázaton ellenőrizhető.)

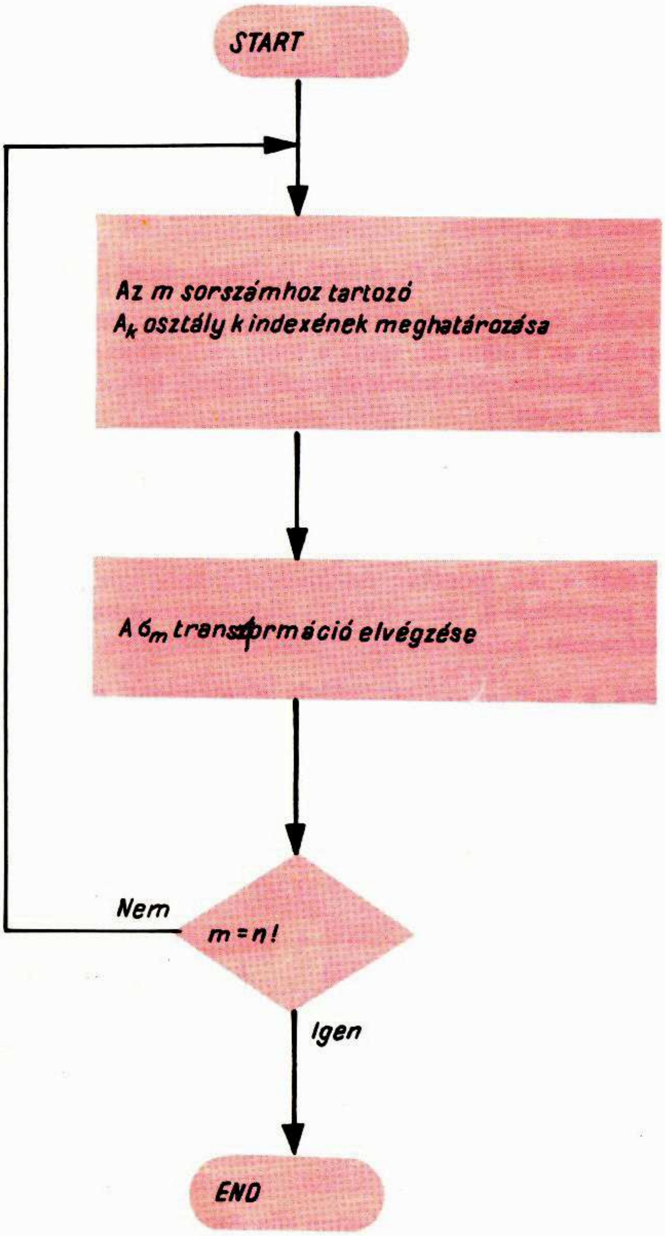
B.4.

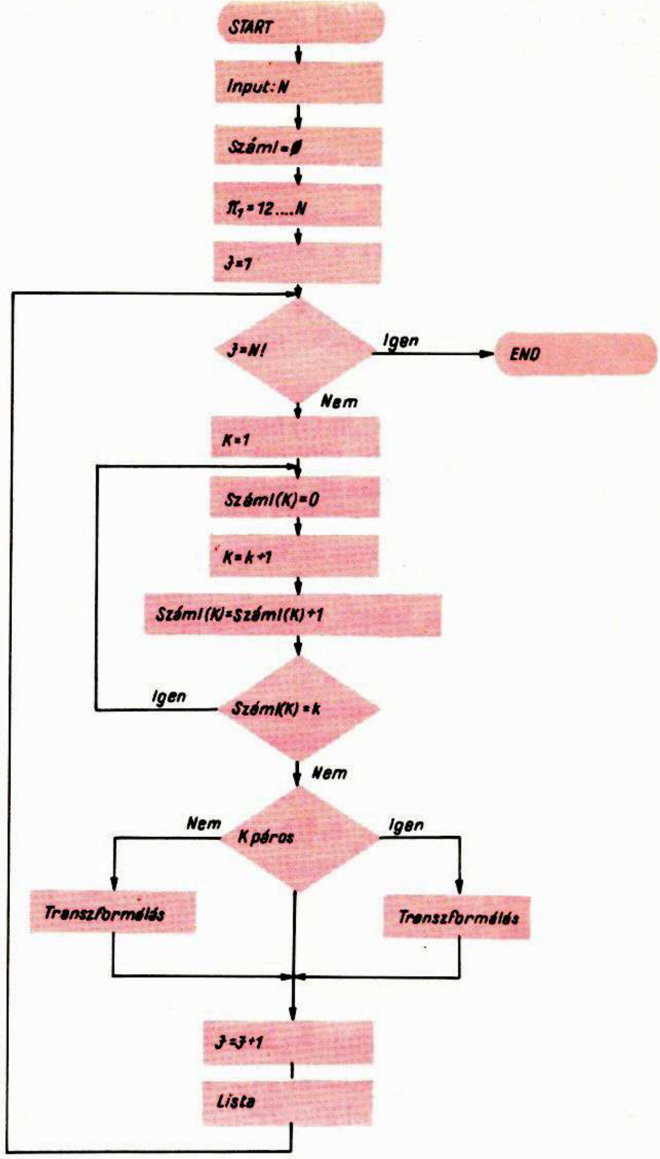
A B.3. pontban adott eljárás fordítottja természetesen egy adott m számhoz rendel hozzá egy permutációt. Itt a lépések pontosan fordított sorrendben történnek, hiszen a b_j sorozat elemei lesznek adva, és ezekből kell az egyes $\ldots \pi'''_m, \pi''_m, \pi'_m, \pi_m$ permutációkat meghatározni, a π_1-ből kiindulva.
Ezzel tehát megadtuk az n-edfokú permutációk és $n!$ db természetes szám közötti kölcsönösen egyértelmű megfeleltetést, amely egy direkt algoritmusnak tekinthető a permutációk előállítására.

A rekurziós eljárásra kidolgozott program és néhány futási eredmény

Az előzőekben ismertetett rekurziós eljárás programvázlatát a *3. ábra* mutatja. Ezen látható, hogy két fő lépésben történik a permutációk előállítása. Az eljárás két n rekeszből álló memóriarészben dolgozik, és ez azt

3. ábra





4. ábra

$$a_1\, a_2 \ldots \underbrace{a_k\, a_{k+1}}_{\tau_k} \ldots a_n$$

5. ábra

jelenti, hogy a számítógép kapacitásához képest nagy n-ek esetén is csak igen csekély mértékben nő a memóriaigény.
Amint a *4. ábra* mutatja, a k osztályindex meghatározása egy egyszerű számlálóban történik, a σ_m transzformációk végrehajtása pedig az elemi transzpozíciók miatt igen egyszerű. Ugyanis a $\tau_k = (k\ k+1)$ transzpozíció nem más, mint a k-adik és $(k+1)$-edik helyen álló elemek felcserélése a π permutációban. ($\pi = a_1 a_2 \cdots a_k a_{k+1} \ldots a_n$ lásd *5. ábra).*
A FORTRAN nyelven írt program egy IBM-360/40 típusú számítógépen futott, és az $n = 3, 4, 5, 6, 7, 8, 9$, értékekre állnak rendelkezésre időeredmények, melyeket a *2. táblázaton* mutatunk be.

2. táblázat

n	$n!$	Előállítási idő
3	6	30 sec
4	24	30 sec
5	120	35 sec
6	720	36 sec
7	5 040	40 sec
8	40 320	79 sec
9	362 880	430 sec

A memóriaszükséglet 1 Kbyte körül mozog, és mint már említettük, a számítógép-memória kapacitásához képest nagyon nagy n szám esetén sem változik jelentősen.
Természetesen az 1 Kbyte memóriaszükségletben a program tárolása is benne van, így a munkaterület ennél jóval kisebb.
Ezt, valamint az algoritmus azon sajátosságát kihasználva, hogy a blokkelső permutációk könnyen meghatározhatók (a közbülső permutációk előállítása nélkül), jelentősen csökkenthető az előállítási idő. Ezt a következőképpen érhetjük el.
Az erre a célra rendelkezésünkre álló memóriától függően, n db $(n-1)$-edfokú blokkelső permutációval ($n(n-1)$ db $(n-2)$-edfokú stb.) párhuzamosan végezzük az előállítást (nn, $n \cdot n(n-1)$, ... stb. rekeszben). Igy a munkaterület csekély növelésével jelentősen csökken az előállítási sebesség.

(Tudományos, műszaki alkalmazások rovat)

Irodalom

[1] DÉNES JÓZSEF: Az összes n-edfokú permutáció előállításáról. Budapest
[2] NEMETZ TIBOR: Permutációk generálása számológépeken. MTA III. Osztály Közleményei, 19. Budapest, 1969.
[3] OBÁDOVICS J. GYULA: Gyakorlati számítási eljárások. Budapest, 1972. Gondolat Kiadó
[4] WELLS, M. B.: Elements of Combinatorial Computing. Pergamon Press, 1971.

A nyugat-európai bankok számítógép-vásárlásai a jövő évtizedben

A Frost and Sullivan Ltd londoni piackutató cég legújabb vizsgálatai szerint az európai bankok az elkövetkezendő tíz évben mintegy 4000 millió font értékben fognak számítógépeket vásárolni.
Ebből az összegből 2500 millió font esik számítógépekre és standard perifériákra, 1500 millió font összeget pedig adatmegjelenítő rendszerekre és speciális, banktevékenységnél használt hardware-re fognak költeni.
A számítógépek felvevő piacának 68%-át Anglia, az NSZK és Franciaország adja; a skandináv országok és a Benelux államok a piac 15%-át jelentik majd.
A tanulmány szerint az 1973. évi 275 millió font értékű berendezéseladás 1982-re majdnem megkétszereződik: 465 millió font értéket fog elérni. Az európai bankok az elkövetkezendő évtizedben számítógépes rendszereik teljes átszervezésére és újra felszerelésére kényszerülnek. A berendezésvásárlás fő tendenciája az írásbeli munkát nem igénylő tranzakciókra és a bankfiók szinten tartott on-line műveletekre való átállást és a terminálok elterjedtebb használatát jelzi.
A tanulmány előrejelzései azt mutatják, hogy ebben az időszakban nemcsak a legnagyobb elektronikus adatfeldolgozó berendezést gyártó cégek fognak jelentős mennyiségű gépet értékesíteni, hanem a kisebb gyárak is képesek lesznek a banktevékenységhez szükséges speciális, terminál-orientált berendezések eladására.
Mivel minden bankban problémát jelent a csekkek hitelességének ellenőrzése, a gyártó cégeknek az 1980-as években jó lehetőségeik lesznek ezt a problémát megoldani képes elektronikus adatfeldolgozó berendezéseik értékesítésére.
Az elektronikus adatfeldolgozás automatizálására fordított tőkebefektetéseket – a piackutató cég előrejelzései szerint – a nyugat-európai bankügyletek évi 10%-os növekedéséből fogják fedezni; a becslések szerint a jelenlegi napi 61 millió tranzakció 1982-re megkétszereződik, azaz eléri a 120 milliót. Egyes országokban, mint pl. Franciaországban az évi növekedés 15%; az NSZK-ban pedig a napi forgalom 24 millió tranzakció lesz, ez a teljes európai forgalom 40%-át jelenti.
A tanulmányban szereplő 16 nyugat-európai ország 331 millió lakosára 40 000 bankszervezet és 1500 felszerelt számítógép jut.
A tanulmány foglalkozik Nyugat-Európa és az Egyesült Államok banktevékenysége közötti legjellemzőbb különbségekkel: az Egyesült Államokban az egyetlen fizetőeszközt ténylegesen a csekkek jelentik. A legtöbb európai országban – Anglia és Franciaország kivételével – a csekk magánszemélyek pénzfelvételét szolgálja, a legtöbb esetben a fizetéseket hitelek és teherátutalások, váltók és más fizetési eszközök révén bonyolítják. Az átutalásoknál használt berendezések nagy része alkalmas on-line adatfeldolgozásra.
A piackutató tanulmány megállapítása szerint a nyugat-európai bankok automatizálása és az elektronikus adatfeldolgozás fejlesztésére vonatkozó tervei nem megnyugtatóak, ha a forgalom várható felfutását vesszük figyelembe.

Data Systems
1974. július–augusztus

Számítástechnikai szolgáltatás bankoknak

Az Egyesült Államokban a bankok egyre inkább követik azt a tendenciát, hogy saját, házon belüli számítástechnikai tevékenység végzése helyett a számítástechnikai szolgáltatásokat veszik igénybe. Ezt bizonyítja az is, hogy a National Bank of Commerce nemrégiben szerződést kötött a University Computing céggel közös számítástechnikai tevékenység végzésére.
A szerződés szerint az University Computing Company az NBC rendelkezésére bocsátja teljes számítógépes programtárát és berendezéseit, amelyek a bank számítástechnikai tevékenységéhez szükségesek. Az NBC cserében olyan műszaki szakembereket és kezelő személyeket biztosít az UCC számára, akik a bankok adatfeldolgozásában nagy tapasztalatokkal rendelkeznek.
A szerződés, melynek értékét 3,6 millió dolláros bevétellel egyenértékűnek tekintik, öt évre szól.

Data Systems
1974. július-augusztus