Kombinatorna prebrojavanja

Bogdan Ljubinković, Miljan Jokić, Dalibor Nikolić, Lazar Jović, Anastazija Petrov, Marko Djordjević, Aleksa Nenadović i Meris Bilalović

Oktobar 2024, FTN

Problemi kojima ćemo se baviti

- Kako se mogu klasifikovati uredjeni izbori elemenata?
- m-premutacije skupa
- Permutacije skupa
- m-permutacije multiskupa
- Permutacije multiskupa

Definicija: Skup

Skup S je kolekcija različitih elemenata, gde svaki element pripada skupu samo jednom.

- Skup se obično zapisuje kao: $S = \{a_1, a_2, \ldots, a_n\}$, gde su a_i jedinstveni elementi.
- U skupu ne postoji redosled elemenata, niti ponavljanje istih.
- Primer: $S = \{1, 2, 3\}$ je skup sa tri elementa.

Definicija: Multiskup

Multiskup M je proširenje pojma skupa gde su ponavljanja elemenata dozvoljena. Formalno, multiskup je funkcija $f : S \to \mathbb{N}$, koja svakom elementu $a \in S$ dodeljuje broj pojavljivanja $f(a)$.

- Multiskup zapisujemo kao: $M = \{\{a_1, a_2, \ldots, a_n\}\}$, gde neki elementi mogu imati višestruka pojavljivanja.
- Redosled elemenata nije bitan, ali broj njihovih pojavljivanja jeste.
- Primer: $M = \{\{1, 1, 2, 3\}\}$ je multiskup sa elementom 1 koji se pojavljuje dva puta.

Uvod u izbore elemenata

Izbor elemenata iz skupa S ili multiskupa M može biti klasifikovan u dve glavne kategorije:

- **Uredjeni izbori**: Redosled elemenata je bitan.
- **Neuredjeni izbori**: Redosled elemenata nije bitan.

Glavni fokus biće na uredjene izbore elemenata.

	m-Permutacije	**Permutacije**
Skup	$\frac{n!}{(n-m)!}$	$n!$
Multiskup	$\frac{n!}{(n-m)! \cdot k_1!k_2!\dots k_r!}$	$\frac{n!}{k_1!k_2!\dots k_r!}$

m-permutacije skupa

Neka je S skup sa n elemenata, i neka je $m \leq n$. m-permutacija skupa S je uredjeni izbor m različitih elemenata iz skupa S.
Broj svih mogućih m-permutacija skupa S izračunava se kao broj injektivnih funkcija $\sigma : \{1, 2, \ldots, m\} \to S$, i dat je formulom:

$$P(n, m) = \frac{n!}{(n - m)!}.$$

- $P(n, m)$ predstavlja broj načina na koji možemo odabrati i urediti m elemenata iz skupa sa n elemenata.
- Redosled elemenata je bitan, ali se svaki element može odabrati najviše jednom.
- Primer: Za skup $S = \{1, 2, 3, 4\}$ i $m = 2$, moguće m-permutacije su $(1, 2), (1, 3), (1, 4), (2, 1), (2, 3), \ldots$

Dokaz: m-permutacije skupa

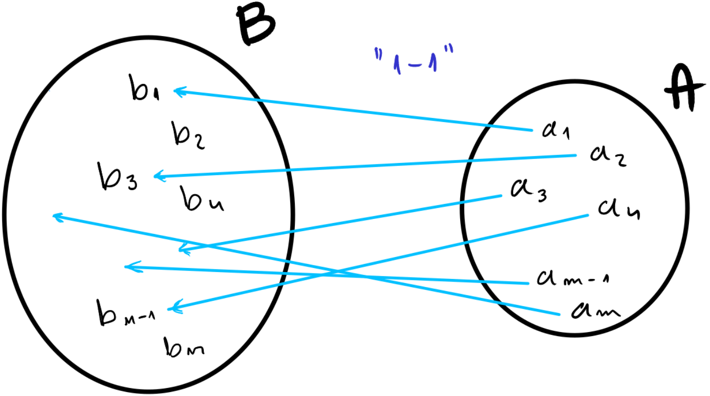
Teza: Broj m-permutacija skupa sa n elemenata je $P(n,m) = \frac{n!}{(n-m)!}$.

Dokaz:

- Prva pozicija: n izbora.
- Druga pozicija: $n-1$ izbora.
- $\vdots$
- m-ta pozicija: $n-m+1$ izbora.

Dakle, broj m-permutacija je:

$$P(n,m) = n \cdot (n-1) \cdots (n-m+1) = \frac{n!}{(n-m)!}.$$


B
"1 – 1"
A
b_1
b_2
b_3
b_4
b_{n-1}
b_n
a_1
a_2
a_3
a_4
a_{m-1}
a_m

Permutacije skupa

Neka je S konačan skup sa n elemenata. Permutacija skupa S je bijekcija $\sigma : S \to S$, što znači da svaka funkcija σ dodeljuje jedinstven element iz S svakom $a \in S$, i svaka vrednost $\sigma(a)$ je različita.
Formalno, σ je takva da:

$$\sigma : S \to S \quad \text{i} \quad \sigma \text{ je bijektivna}, \quad \sigma(a_i) \neq \sigma(a_j) \quad \text{za} \quad i \neq j.$$

Broj svih permutacija skupa sa n elemenata jednak je broju bijekcija od S na sebe, tj. dat je sa $n!$:

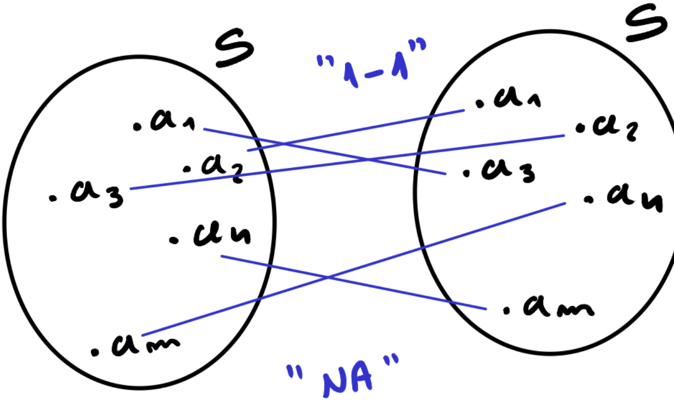
$$P(n) = |\mathrm{Sym}(S)| = n!.$$

Dokaz: Permutacije skupa

Teza: Broj permutacija skupa sa n elemenata je $P(n) = n!$.
Dokaz:

- Za $m = n$, imamo $P(n, n)$, što znači da biramo svih n elemenata i uredjujemo ih.
- Prema formuli za m-permutacije, imamo:

$$P(n, n) = \frac{n!}{(n-n)!} = \frac{n!}{0!} = n!.$$



S
S
"1-1"
.a1
.a2
.a3
.an
.am
.a1
.a2
.a3
.an
.am
"NA"

m-permutacije multiskupa

Neka je M multiskup sa n elemenata, gde neki elementi mogu biti ponovljeni. m-permutacija multiskupa M predstavlja uredjeni izbor m elemenata iz M.

Ako se element a_i pojavljuje k_i puta, broj m-permutacija dat je formulom:

$$P(M, m) = \frac{n!}{(n-m)! \cdot k_1! k_2! \dots k_r!},$$

gde su $k_1, k_2, \dots, k_r$ multipliciteti elemenata koje biramo.

- Na primer, za multiskup $M = \{\{a, a, b, b, c\}\}$ i $m = 3$, broj m-permutacija uzimajući u obzir multiplicitete se izračunava prema datoj formuli.

Dokaz: m-permutacije multiskupa (1/2)

Teza: Broj m-permutacija multiskupa M sa n elemenata, gde su neki elementi višestruki, je

$$P(M, m) = \frac{n!}{(n-m)! \cdot k_1! \cdots k_r!}.$$

Prvi korak: Biranje m elemenata iz n:

- Odabiremo m elemenata iz n, uzimajući u obzir višestruke instance.

Dokaz: m-permutacije multiskupa (2/2)

Drugi korak: Uredjivanje odabranih elemenata:

- Uredjujemo m elemenata, što daje $m!$ redosleda.
- Da bismo isključili duplikate, delimo sa $k_1! \cdots k_r!$ (broj permutacija multipliciteta), gde su k_i multipliciteti pojedinih elemenata.

Konačno, broj m-permutacija multiskupa postaje:

$$P(M, m) = \frac{n!}{(n-m)! \cdot k_1! \cdots k_r!}.$$

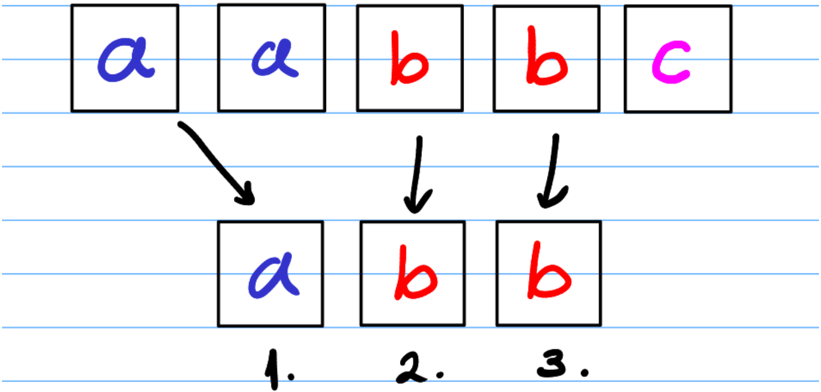


Figure: Primer jedne permutacije

Permutacije multiskupa

Neka je M multiskup sa n elemenata, gde neki elementi mogu biti ponovljeni. Permutacija multiskupa M predstavlja uredjeni raspored svih njegovih elemenata.
Ako se element a_i pojavljuje k_i puta, broj permutacija dat je formulom:

$$P(M) = \frac{n!}{k_1!k_2!\dots k_r!},$$

gde su $k_1, k_2, \dots, k_r$ multipliciteti elemenata, a $n = k_1 + k_2 + \cdots + k_r$.

- Na primer, za multiskup $M = \{a, a, b\}$, broj permutacija je:

$$P(M) = \frac{3!}{2!1!} = 3.$$

Dokaz: Permutacije multiskupa (1/2)

Teza: Broj permutacija multiskupa M sa n elemenata, gde neki elementi mogu biti višestruki, je:

$$P(M) = \frac{n!}{k_1! \cdots k_r!}.$$

Razmatranje: Kada uredjujemo n elemenata, svaki od njih može biti isti kao i drugi.

- Neka su k_i multipliciteti elemenata u multiskupu.
- $n = k_1 + k_2 + \ldots + k_r$ je ukupan broj elemenata.

Dokaz: Permutacije multiskupa (2/2)

Koraci:

- Broj svih rasporeda n elemenata je $n!$.
- Duplikati se pojavljuju zbog višestrukih instanci, pa delimo sa $k_1! \cdots k_r!$.
- Tako dobijamo konačnu formulu:

$$P(M) = \frac{n!}{k_1! \cdots k_r!}.$$

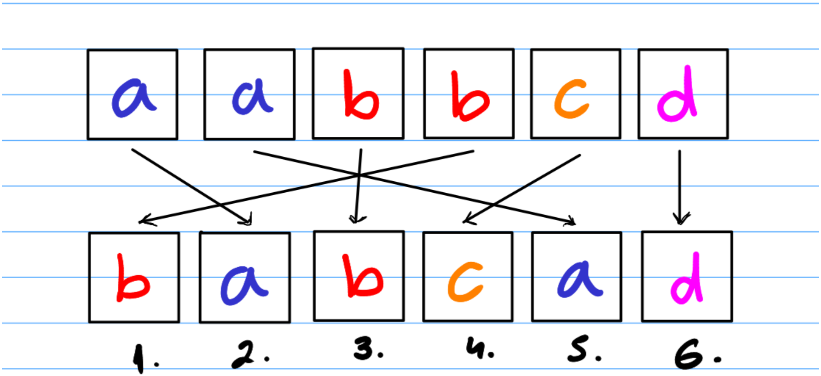


Figure: Primer jedne permutacije

Zadatak 1

Neka je data abeceda od $n = 26$ različitih slova (A-Z). Koliko različitih "reči" (nizova) dužine $m = 5$ možemo formirati ako:

- Prva dva slova moraju biti samoglasnici (A, E, I, O, U).
- Preostala tri slova moraju biti suglasnici.
- Svako slovo može biti korišćeno samo jednom.

Rešenje - Korak 1 i 2

1. Izbor samoglasnika:
Prva dva slova mogu biti od 5 samoglasnika. Odabiremo 2 od 5 samoglasnika:

$$P(5,2) = \frac{5!}{(5-2)!} = 5 \times 4 = 20$$

2. Izbor suglasnika:
U abecedi ima 21 suglasnik. Treba nam 3 suglasnika:

$$P(21,3) = \frac{21!}{(21-3)!} = 21 \times 20 \times 19 = 7980$$

Rešenje - Korak 3

3. Kombinovanje samoglasnika i suglasnika:

$$\text{Ukupno reči} = P(5,2) \times P(21,3) = 20 \times 7980 = 159600$$

Dakle, ukupno možemo formirati **159600** različitih "reči".

Zadatak 2

Bogdan želi da jede sladoled, ali nije siguran koji izbor ukusa je najbolji. Odrediti sve mogućnosti da Bogdan odabere kugle ukoliko postoji 9 ukusa i važe sledeće limitacije:

- Sve kugle moraju biti različite.
- Redosled kugli je bitan zbog uklapanja ukusa.
- Mora se odabrati bar 1, a maksimalno 3 kugle (na dieti je).

Rešenje - Korak 1 i 2

1. **Izbor 1 kugle:**
Za slučaj odabira samo 1 kugle broj opcija je očigledan:

$$P(1, 9) = \frac{9!}{(9-1)!} = 9$$

2. **Izbor 2 kugle:**
Isto razmišljanje je i za 2 kugle:

$$P(2, 9) = \frac{9!}{(9-2)!} = 9 \times 8 = 72$$

Rešenje - Korak 3 i 4

3. Izbor 3 kugle:

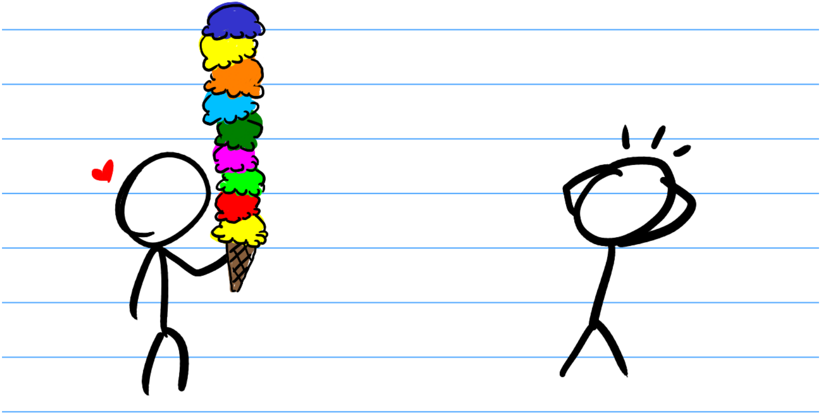
$$P(3,9) = \frac{9!}{(9-3)!} = 9 \times 8 \times 7 = 504$$

4. Ukupan rezultat:

$$9 + 72 + 504 = 584$$

Napomena:
Ako bismo posmatrali zavisnost ovog slučaja od **n** kao broja ukusa i **m** kao maksimalan broj kugli rešenje bi se moglo definisati kao:

$$\sum_{i=1}^{m} P(i,n)$$



```python
def generisi_permutacije_skupa(A):
    if len(A) == 0:
        return [[]]

    permutacije = []

    for i in range(len(A)):
        trenutni_elem = A[i]

        ostatak = A[:i] + A[i + 1:]

        # Rekurzivno generisanje permutacija za ostale elemente skupa
        for p in generisi_permutacije_skupa(ostatak):
            # Dodavanje trenutnog elementa na permutacije ostatka skupa
            permutacije.append([trenutni_elem] + p)

    return permutacije
```

Figure: Programerski zadatak - permutacije skupa

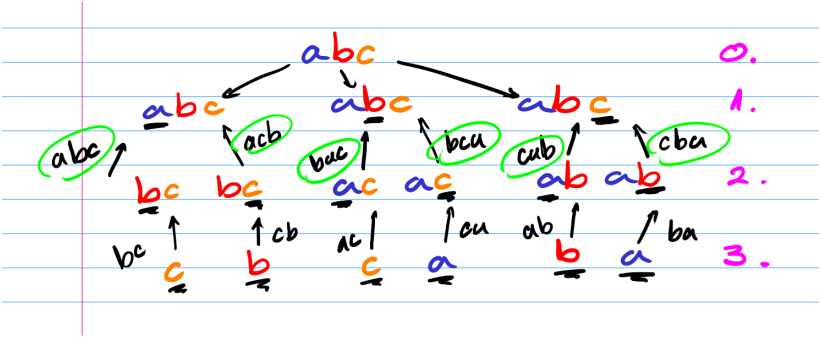


Figure: Stablo razvoja - u svakom nivou "pinujemo" jedan element

```python
def generisi_permutacije_multiskupa(A):
    A.sort()

    if len(A) == 0:
        return [[]]

    permutacije = []

    for i in range(len(A)):
        if i > 0 and A[i] == A[i - 1]:
            continue

        current = A[i]

        ostatak = A[:i] + A[i+1:]

        # Rekurzivno generisanje permutacija za ostale elemente multiskupa
        for p in generisi_permutacije_multiskupa(ostatak):
            # Dodavanje trenutnog elementa na permutacije ostatka multiskupa
            permutacije.append([current] + p)

    return permutacije
```

Figure: Programerski zadatak - permutacije multiskupa

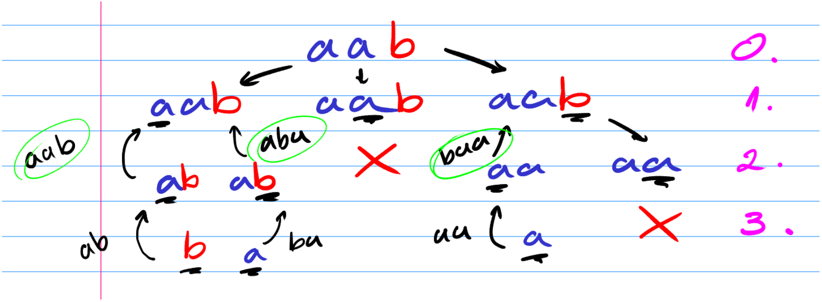


Figure: Ako je element isti kao prethodni - ne razvijamo stablo dalje

```python
def generisi_m_permutacije_skupa(A, m):

    A.sort()

    if m == 0:
        return [[]]

    mpermutacije = []

    for i in range(len(A)):

        current = A[i]

        ostatak = A[:i] + A[i+1:]

        # Rekurzivno generisanje (m - 1) permutacija za ostale elemente skupa
        for p in generisi_m_permutacije_skupa(ostatak, m - 1):
            # Dodavanje trenutnog elementa na permutacije ostatka skupa
            mpermutacije.append([current] + p)

    return mpermutacije
```

Figure: Programerski zadatak - m-permutacije skupa

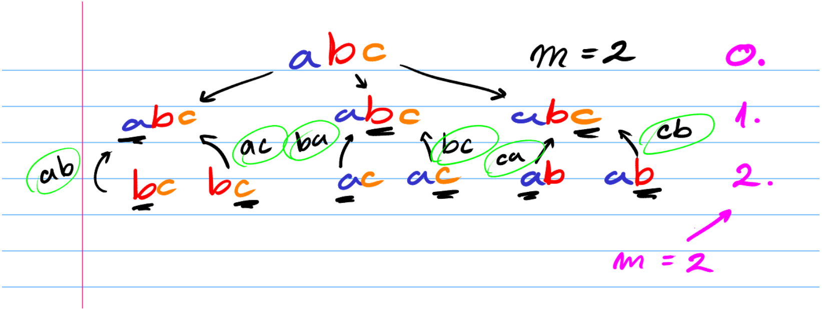


Figure: Razvijamo do onog nivoa koliko je m

```python
def generisi_m_permutacije_multiskupa(A, m):

    A.sort()

    if m == 0:
        return [[]]

    mpermutacije = []

    for i in range(len(A)):
        if i > 0 and A[i] == A[i - 1]:
            continue

        current = A[i]

        ostatak = A[:i] + A[i+1:]

        # Rekurzivno generisanje (m - 1) permutacija za ostale elemente multiskupa
        for p in generisi_m_permutacije_multiskupa(ostatak, m - 1):
            # Dodavanje trenutnog elementa na permutacije ostatka multiskupa
            mpermutacije.append([current] + p)

    return mpermutacije
```

Figure: Programerski zadatak - m-permutacije multiskupa

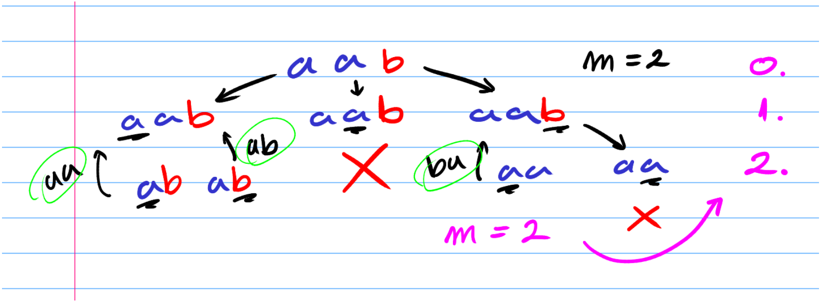


Figure: Razvijamo onoliko nivoa koliko je m, preskačemo ako smo korak ranije već videli element