МІНІСТЕРСТВО ОСВІТИ І НАУКИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»

Кафедра систем штучного інтелекту

Лабораторна робота №6

З курсу “Дискретна математика ”

Виконав:
ст.гр. КН-110
Холод Ігор
Викладач:
Мельникова Н.І.

Львів – 2018

Лабораторна робота № 6.

В.13

Тема: Генерація комбінаторних конфігурацій

Мета роботи: набути практичних вмінь та навичок при комп'ютерній реалізації комбінаторних задач.

Теоретичні відомості

Головна задача комбінаторики – підрахунок та перелік елементів у скінчених множинах.
Правило додавання: якщо елемент – x може бути вибрано n способами, а y- іншими m способами, тоді вибір „ x або y" може бути здійснено (m+n) способами.
Правило добутку: якщо елемент – x може бути вибрано n способами, після чого y - m способами, тоді вибір упорядкованої пари (x, y) може бути здійснено (m*n) способами.
Набір елементів $xi1$, $xi2$, ..., xim з множини $X = \{x1, x2, ..., xn\}$ називається вибіркою об'єму m з n
елементів – *(n, m)* – **вибіркою**.
Упорядкована (n, m) – вибірка, в якій елементи не можуть повторюватися, називається *(n, m)* –
розміщеням, кількість всіх можливих розміщень обчислюється за формулою:

$$A_n^m = \frac{n!}{(m-n)!}$$

Упорядкована (n, m) – вибірка, в якій елементи можуть повторюватися, називається *(n, m)* –
розміщеням з повторюваннями, кількість всіх можливих таких розміщень обчислюється за формулою:

$$\overline{A_n^m} = n^m$$

Неупорядкована (n, m) – вибірка, в якій елементи не можуть повторюватися, називається *(n,*
m) – **сполученням**, кількість всіх можливих сполучень обчислюється за формулою:

$$C_n^m = \frac{n!}{(n-m)!\,m!}$$

Неупорядкована (n, m) – вибірка, в якій елементи можуть повторюватися, називається *(n,m)-*

сполученням з повторюваннями, кількість всіх можливих таких сполучень обчислюється за формулою:

$$\overline{\mathrm{C}_{\mathrm{n}}^{\mathrm{m}}} = \mathrm{C}_{\mathrm{m+n-1}}^{\mathrm{m}}$$

A_n^n – називається **перестановкою**, а кількість різних перестановок позначається та обчислюється за формулою:

$$P_n = n!$$

Якщо в перестановках є однакові елементи, а саме перший елемент присутній *n*1 разів, другий елемент – n_2 разів, ... , k-ий елемент – n_k разів, причому $n_1 + n_2 + \cdots + n_k = n$ то їх називають **перестановками з повторенням** та кількість їх можна знайти за формулою:

$$P(n_1, n_2, \dots, n_k) = \frac{n!}{n_1!\, n_2! \dots n_k!}$$

Розв'язання

1. Чоловік протягом 14 днів мати був прочитати 14 журналів, причому в день він читав лише один журнал. Скількома варіантами він міг прочитати всі журнали?

У перший день чоловік міг обрати один з 14 журналів, у другий один з 13, у третій – один з 12 і т.д. Тому загальна кількість можливих комбінацій **14!**

2. Скільки різних трицифрових натуральних чисел можна скласти з цифр 1, 2, 3, 4, 5 за умови, що в кожне число входить цифра не більше одного разу?

Перестановка трьох чисел, які ми обираємо з загальної множини чисел, потужність якої 5, при чому порядок має значення – це розміщення без повторень:

$$A_5^3 = \frac{5!}{2!} = 60$$

3. Скількома способами можна вибрати трьох чергових із класу, в якому навчається 20 учнів?

З 20 учнів ми обираємо 3, при чому в якому порядку ми їх оберемо не має значення, тому це – сполучення без повторень:

$$C_{20}^3 = \frac{20!}{17!\, 3!} = 1140$$

4. Скількома способами можна розділити 6 різних іграшок та 5 різних книжок між 3 дітьми?

Перша іграшка віддається одній з трьох дітей. Друга незалежно від цього віддається одній з трьох дітей і т.д. Так само і для книг. Тобто це – розміщення з повторенням:

$$\overline{A_6^3} = 6^3 = 216 \text{ способів роздати іграшки}$$

$$\overline{A_5^3} = 5^3 = 125 \text{ способів роздати книги}$$

Загалом отримаємо 216*125 = 27 000 способів

5. Скількома способами можна поділити 9 однакових яблук та 6 однакових груш між трьома чоловіками?

Міркування такі ж, як і у попередній задачі, але тепер порядок значення не має, тому використовуємо сполучення з повтореннями:

$$\overline{C_9^3} * \overline{C_6^3} = C_{11}^3 * C_8^3 = 9240 \text{ способів}$$

6. П'ять учнів вирішили написати всі необхідні 15 білетів, яки пропонував викладач на екзамен з філософії. При цьому кількість написаних кожним з них білетів розподілили так – перший має написати 4 білета, другий – 3, третій – 2, четвертий – 1, п'ятий – 5. Скількома способами можна розподілити таким чином всі білети між ними?

Перший студент обирає чотири білета з 15, другий обирає 3 білета, але вже з 11, третій – два з 8, четвертий – один з 6, і п'ятий – п'ять з 5.

$$C_{15}^4 * C_{11}^3 * C_8^2 * C_6^1 * C_5^5 = 37\,837\,800$$

7. Скільки чотирьохзначних чисел діляться хоча б на одне з чисел 12, 8?

Нехай A = множина чотирьохзначних чисел, які діляться на 8, B – на 12.
|A| = 1125, |B| = 750, $|A \cap B|$ = 375.
Тоді за формулою включень виключень:

$$|A \cup B| = |A| + |B| - |A \cap B| = 1500$$

Частина 2. Запрограмувати за варіантом обчислення кількості розміщення(перестановок, комбінацій, алгоритму визначення наступної лексикографічної сполуки, перестановки) та формули Ньютона і побудувати за допомогою неї розклад за варіантом.
Задане додатне ціле число n і невід'ємне ціле число r (r <= n). Розташувати у лексикографічному порядку всі розміщення без повторень із елементів множини {1, 2, …, n}. Побудувати розклад $(x - y)^{11}$

Код програми:

```
#include <stdio.h>

//factorial (recursion)
unsigned long long fact(unsigned int n)
{
    if (n >= 1)
        return n*fact(n - 1);
    else
        return 1;
}

//counting binomial coefficient
int coef(int n, int k)
{
    return ( fact(n)/(fact(k)*fact(n-k)) );
}

//printing binom (x - y) ^ _pow
void minus_binom(int _pow)
{
    int k = 0, n = _pow;
    printf("Your binom:\n");
    printf("(x-y)^%i = ", n);
    for ( ; k <= _pow; k++)
    {
        if (k == 0)
            printf("x^%i + ", n);
        else if (k == n)
            printf("y^%i\n", k);
        else
        {
            if (k % 2 != 0) //when k-power is odd number
                printf("%i*x^%i*y^%i - ", coef(n, k), n - k, k);
            else //when k-power is even number
                printf("%i*x^%i*y^%i + ", coef(n, k), n - k, k);
        }
    }
}

void swap(int *arr, int i, int j) //swapping 2 elements
{
    int temp = arr[i];
    arr[i] = arr[j];
    arr[j] = temp;
}

int Next_A(int *a, int n, int m) //searching for the next permutation
{
    int left; //left number

    //while next permutation is not found yet
    //in the end, left number has to be in boards (0; m - 1)
```

```
    // (-1) because arrays indexes start with 0
    do
    {
        left = n - 1;
        //going from the right to the left
        //and searching hindmost(крайній) left number,
        //so that number to the right is lower
        while (left != -1 && a[left] >= a[left + 1])
            left--;
        if (left == -1)
            return 0; //last permutation has already found
        int right = n - 1;
        //searching nearest (to j-th) element that is higher
        while (a[left] >= a[right])
            right--;
        swap(a, left, right); //swapping j-th and j-th elements
        //sorting all elements to the right of j-th
        //as in the last iteration elements, starting from j-th
        //are in descending order
        //its enough to alternately swap places of last and first elements after j-th
        int l = left + 1, r = n - 1;
        while (l < r)
            swap(a, l++, r--);
    } while (left > m - 1);
    return 1;
}

void Print_A(int *a, int m)  //printing a permutation
{
    static int num = 1; //number of permutation
    printf("%i:   ", num++);
    for (int i = 0; i < m; i++)
        printf("%i ", a[i]);
    printf("\n");
}

int main()
{
    int pow;
    printf("Enter a power of binom: ");
    scanf("%d", &pow);
    minus_binom(pow);

    printf("\nInput n and r:\n");
    int n, r;
    printf("n = ");
    scanf("%d", &n);
    printf("r = ");
    scanf("%d", &r);
    int a[n];
    //intializating array a as {1, 2, 3, ...}
    for (int i = 0; i < n; i++)
        a[i] = i + 1;
    Print_A(a, r); //printing first permutation
```

```
    
    //printing next permutations
    while (Next_A(a, n, r))
        Print_A(a, r);
    return 0;
}
```

Програма складається з двох частин. Перша частина шукає розклад $(x-y)^n$, де n – степінь, що вводиться користувачем. Друга частина виводить розміщення виводить розміщення A_n^r в лексикографічному порядку, де n і r – числа, які вводяться користовучем.

Висновок: я освоїм основні знання з теорії комбінаторики, а також навчився програмно реалізовувати деякі задачі з комбінаторики.