МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ НАЦІОНАЛЬНОМУ УНІВЕРСИТЕТІ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”

Кафедра систем штучного інтелекту

Лабораторна робота №6 з дисципліни «Дискретна математика»

Виконав:
студент групи КН-114
Пилипів Андрій
Викладач:
Мельникова Н.І.

Львів – 2019 р

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

ТЕОРЕТИЧНІ ВІДОМОСТІ ТА ПРИКЛАДИ РОЗВ'ЯЗАННЯ ЗАДАЧ

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:

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

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:

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

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

$$\overline{C_n^m} = C_{n+m-1}^m$$

– називається перестановкою, а кількість різних перестановок позначається та обчислюється за формулою:

$$Pn = n!$$

. Якщо в перестановках є однакові елементи, а саме перший елемент присутній n1 разів, другий елемент – n2 разів, ... , k-ий елемент – nk разів, причому n1+n2 +....+nk =n , то їх називають перестановками з повторенням та кількість їх можна знайти за формулою

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

Варіант № 6

1. Скільки різних бус можна зробити з 15 різних бусинок?
2. Скільки різних трицифрових натуральних чисел можна скласти з цифр 1, 2, 3, 4, 5, щоб у ньому кожна з цих цифр зустрічалась не більше одного разу?
3. З лабораторії, у якій працює 25 чоловік, 5 співробітників мають поїхати у відрядження. Скільки може бути різних складів цієї групи?
4. Із 12 тенісистів і 6 тенісисток формують три змішані пари (до пари входять по одному тенісисту й одній тенісистці). Скількома способами це можна зробити?
5. На книжковій полиці вміщується тринадцять томів енциклопедії. Скількома способами їх можна розставити так, щоб томи 1 і 2 стояли поруч?
6. У турнірі беруть участь 12 шахістів. Визначити кількість різних розкладів першого туру (розклади вважаються різними, якщо вони відрізняються учасниками; колір та номер столу не враховується)
7. Знайти кількість цілих додатних чисел, що не більше 9000 і не діляться на жодне з чисел 12, 36 і 52.

1) P15 = 15!
2) 5*4*3
3) C(5,25)
4) 12*6+11*5+10*4
5) P12 = 12!
6) C(2,12)
7)

```
int count = 0;
for (int i = 1; i <= 9000; i++)if (i % 12 != 0 && i % 36 != 0 && i % 52 != 0)count++;
cout << "Count = " << count;
```

```
Count = 8134
```

Завдання №2. Запрограмувати за варіантом обчислення кількості розміщення(перестановок, комбінацій, алгоритму визначення наступної лексикографічної сполуки, перестановки) та формулу Ньютона і побудувати за допомогою неї розклад за варіантом

```cpp
int number = 1;//counter

bool next(int* a, int n, int m)
{
    int index = m - 1;
    while (a[index] == n && index >= 0) index--;
    if (index < 0) return false;
    if (a[index] >= n)index--;
    a[index]++;
    if (index == m - 1) return true;
    for (int k = index + 1; k < m; k++)a[k] = a[index];
    return true;
}
void show(int* a, int n)
{
    cout << setw(6) << number++ << "|";
    for (int i = 0; i < n; i++)
        cout << a[i] << " ";
    cout << endl;
}
void main()
{
    int n, m, * a;
    cout << "Max number = ";cin >> n;
    cout << "Count = ";cin >> m;
    a = new int[n];
    for (int i = 0; i < n; i++)
        a[i] = 1;
    cout << setw(6) << "Number" << "|" << "Set" << endl;
    show(a, m);
    while (next(a, n, m))
        show(a, m);}
```

```
Max number = 5
Count = 3
Number|Set
     1|1 1 1
     2|1 1 2
     3|1 1 3
     4|1 1 4
     5|1 1 5
     6|1 2 2
     7|1 2 3
     8|1 2 4
     9|1 2 5
    10|1 3 3
    11|1 3 4
    12|1 3 5
    13|1 4 4
    14|1 4 5
    15|1 5 5
    16|2 2 2
    17|2 2 3
    18|2 2 4
    19|2 2 5
    20|2 3 3
    21|2 3 4
    22|2 3 5
    23|2 4 4
    24|2 4 5
    25|2 5 5
    26|3 3 3
    27|3 3 4
    28|3 3 5
    29|3 4 4
    30|3 4 5
    31|3 5 5
    32|4 4 4
    33|4 4 5
    34|4 5 5
    35|5 5 5
```

Біном Ньютона

```cpp
long double factorial(int n){return n > 0 ? n * factorial(n - 1) : 1;}

double binome(int a, int b, int n) {
    double sum = 0;
    for (int k = 0; k <= n; k++)
        sum += (factorial(n) / (factorial(k) * factorial(n - k))) * pow(a, k) * pow(b, n - k);
    return sum;
}

int main()
{
    int a, b, n;
    cout << "a = ";cin >> a;
    cout << "b = ";cin >> b;
    cout << "n = ";cin >> n;
    cout << "("<<a<<" + "<<b<<")^"<<n<<" = "<<binome(a, b, n);
    return 0;
}
```

```
a = 2
b = 4
n = 5
(2 + 4)^5 = 7776
```