

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ НАЦІОНАЛЬНОМУ
УНІВЕРСИТЕТІ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”

Кафедра систем штучного інтелекту

Лабораторна робота № 6
з дисципліни “дискретна математика”

Виконав:

студент групи КН-115

Манчур Іван

Викладач:

Мельникова Н.І.

львів - 2019 рік

Тема: Генерація комбінаторних конфігурацій

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

Варіант - 4

1. Скількома способами можна видати 15 учням:

а) 15 різних варіантів білетів;

б) 5 білетів першого варіанта, 5 – другого, 5 – третього?

а) Враховуючи, що всі білети різні, маємо розміщення з 15 білетів на 15 учнів, тобто $15!$ способів.

б) Три варіанти повторюються, тому буде розміщення з повторенням, де $N_1 = 5, N_2 = 5, N_3 = 5$

тому $P_{15}(5,5,5) = (15!)/(3 \cdot (5!)) = 756756$

2. Скількома способами можна розділити 6 різних цукерок між трьома дітьми?

$$C_6^3 = 6! / (3! \cdot 3!) = 20$$

3. Скількома способами можна розташувати 12 різних деталей у трьох однакових ящиках?

Маємо розташування без повторень (різні деталі), отже

$$A_{12}^3 = 12!/9! = 1320$$

4. Збори, на яких присутні 40 чоловік, обирають голову, секретаря і трьох членів комісії. Скількома способами це можна зробити?

Є C_{40}^1 способів обрати голову,
 C_{39}^1 обрати секретаря і
 C_{38}^3 обрати 3 членів комісії

За правилом добутку отримуємо $C_{40}^1 * C_{39}^1 * C_{38}^3$
 способів ,тобто 13160160 способів

5. Для учнів класу було куплено 20 білетів у театр на місцях, що знаходяться в одному ряду (на якому 20 місць). Скільки є способів розподілу цих білетів між учнями (10 хлопців та 10 дівчат), щоб два хлопця або дві дівчини не сиділи поруч?

Розв'язання: Спочатку розсадимо дівчат. Це можливо зробити 10! способами. Тепер розсадимо хлопців, Усього можливих місць маємо 10, тому кількість способів розсадження дорівнює 10! також. Так як дівчата ,можуть сидіти на місцях хлопців і навпаки ,то необхідно результат домножити на 2

,отже маємо $10! * 10! * 2$ способів.

6. Десятьох тенісистів мають розподілити на групи по 2, 3 і 5 спортсменів для поїздки на три турніри, які обираються з 6 можливих. Скількома способами це можна зробити?

Для початку виберемо 3 турніри з шести можливих. Це можливо зробити 6A_3 способами. Тепер розіб'ємо тенісистів на 3 турніри, це буде упорядковане розбиття, тобто маємо $N(0,1,1,1,0,0,0,0,0,0) = 10!/1!*1!*1!*(2!)^1*(3!)^1*(5!)^1 = 3628800/1440 = 2520$.

Тоді за правилом добутку отримаємо $216*2520 = 544320$ способів.

7. Знайдіть кількість цілих додатних чисел, що не більше 1000 і не діляться на жодне з чисел 3, 5 і 7.

Відповідь: 457 цілих чисел не більші 1000 і не діляться на жодне з чисел 3, 5, 7

Додаток 2

Задане додатне ціле число n . Побудувати всі сполуки без повторень елементів множини $\{1, 2, \dots, n\}$. Побудувати розклад $(a - b)^6$

```

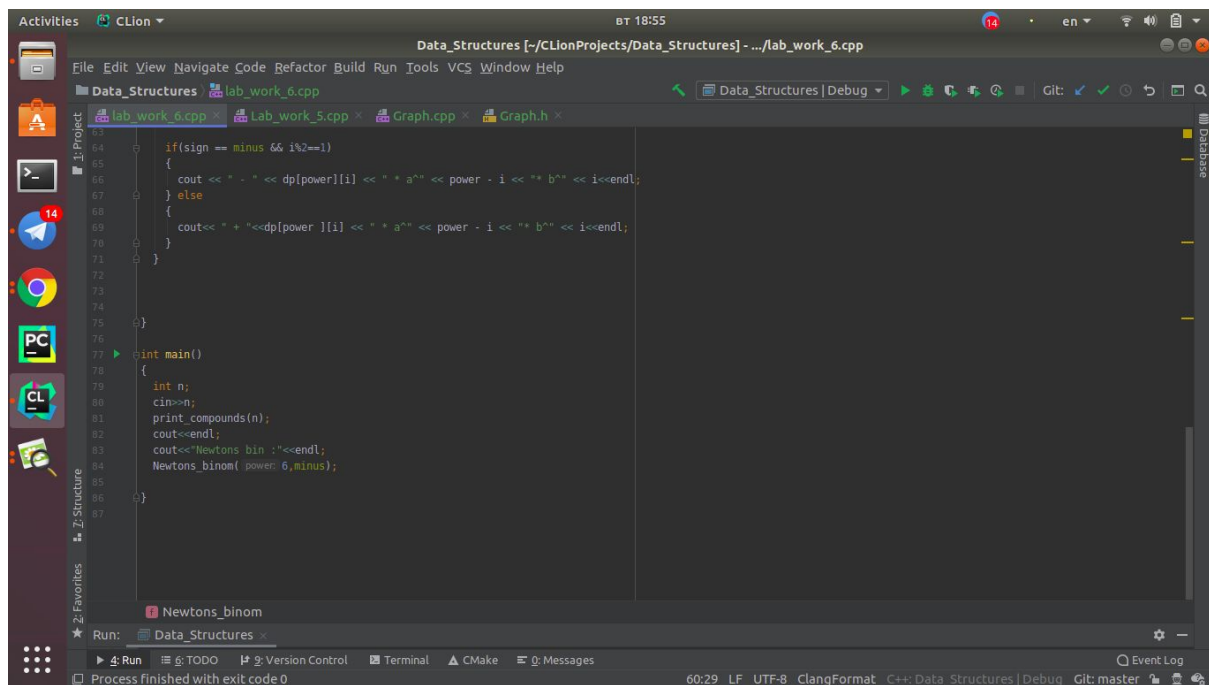
1 // Created by Ivan on 01.12.19.
2 //
3 #include <cassert>
4 #include <iostream>
5 #include <cmath>
6 #define minus false
7 #define plus true
8 using namespace std;
9
10
11
12 // prints from 1 to n
13 void print_compounds(int n)
14 {
15     int cur_pos;
16     for(int i = 0; i < pow(2, n); i++) { ... }
17 }
18
19
20 int dp[100][100];
21 int f(int i, int j)
22 {
23     if(dp[i][j] != 0)
24     {
25         return dp[i][j];
26     }
27     else if(i == 0)
28     {
29         return 1;
30     }
31     else if(i == j)
32     {
33         return 1;
34     }
35     else
36     {
37         return dp[i][j] = f(i - 1, j) + f(i - 1, j - 1);
38     }
39 }
40
41 void Newtons_binom(int power, bool sign = plus)
42 {
43     for(int i = 0; i <= power; i++)
44     {
45         f(power, i);
46         if(sign == minus && i % 2 == 1)
47         {
48             cout << " - " << dp[power][i] << " * a" << power - i << " * b" << i << endl;
49         }
50         else
51         {
52             cout << " + " << dp[power][i] << " * a" << power - i << " * b" << i << endl;
53         }
54     }
55 }
56
57 int main()
58 {
59     int n;
60     cin >> n;
61     print_compounds(n);
62     Newtons_binom(n);
63     return 0;
64 }

```

```

42     return dp[i][j];
43 }
44 else if(j == 0)
45 {
46     return dp[i][j] = 1;
47 }
48 else if(i == j)
49 {
50     return dp[i][j] = 1;
51 }
52 else
53 {
54     return dp[i][j] = f(i - 1, j) + f(i - 1, j - 1);
55 }
56 }
57
58 void Newtons_binom(int power, bool sign = plus)
59 {
60     for(int i = 0; i <= power; i++)
61     {
62         f(power, i);
63         if(sign == minus && i % 2 == 1)
64         {
65             cout << " - " << dp[power][i] << " * a" << power - i << " * b" << i << endl;
66         }
67         else
68         {
69             cout << " + " << dp[power][i] << " * a" << power - i << " * b" << i << endl;
70         }
71     }
72 }
73
74 int main()
75 {
76     int n;
77     cin >> n;
78     print_compounds(n);
79     Newtons_binom(n);
80     return 0;
81 }

```



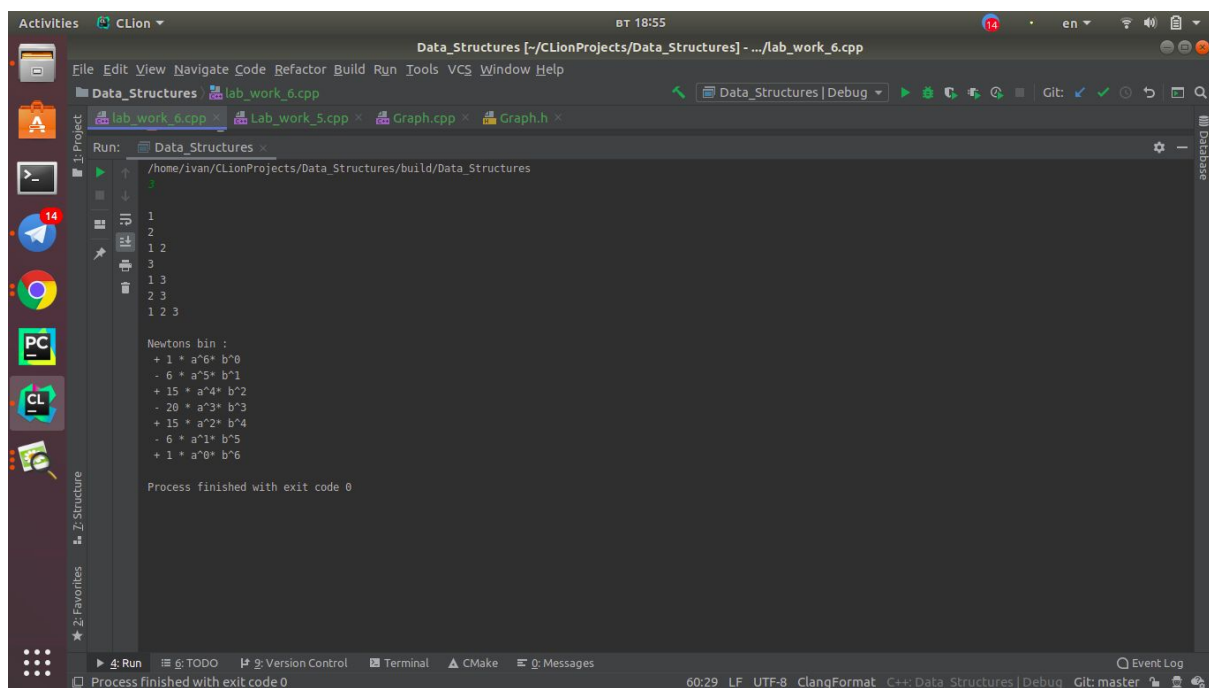
```
1: Project
Data_Structures
lab_work_6.cpp
lab_work_5.cpp
Graph.cpp
Graph.h

if(sign == minus && i%2==1)
{
    cout << " - " << dp[power][i] << " * a" << power - i << " * b" << i<<endl;
} else
{
    cout << " + " << dp[power][i] << " * a" << power - i << " * b" << i<<endl;
}

int main()
{
    int n;
    cin >> n;
    print_compounds(n);
    cout << endl;
    cout << "Newtons bin : " << endl;
    Newtons_binom( power, minus);
}
```

Run: Data_Structures

Process finished with exit code 0



```
Run: Data_Structures
/home/ivan/CLionProjects/Data_Structures/build/Data_Structures
1
2
1 2
3
1 3
2 3
1 2 3

Newtons bin :
+ 1 * a^6 * b^0
- 6 * a^5 * b^1
+ 15 * a^4 * b^2
- 20 * a^3 * b^3
+ 15 * a^2 * b^4
- 6 * a^1 * b^5
+ 1 * a^0 * b^6

Process finished with exit code 0
```

Run: Data_Structures

Process finished with exit code 0

Висновок: на цій практичній роботі я набув практичних умінь та навичок при комп'ютерній реалізації комбінаторних задач