РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ**

**ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ им. В. Г. ШУХОВА»**

**(БГТУ им. В.Г. Шухова)**

Кафедра программного обеспечения вычислительной техники и автоматизированных систем

**Лабораторная работа №2.2**

по дисциплине: Дискретная математика

тема: «Задачи выбора»

Выполнил: ст. группы ПВ-211

Медведев Дмитрий Сергеевич

Проверили:

Рязанов Юрий Дмитриевич

Белгород 2022 г.

**Цель занятия:** научиться решать задачи выбора.

**Задача:** Найти все решения уравнения x1 + x2 + ... + xn = b в натуральных числах, n и b — заданы.

**Комбинаторный объект:** размещение с повторением.

Функционал: размещения n натуральных чисел меньше искомого b.

Способ распознавания решения по значению функционала: получить сумму элементов полученного множества, если оно равно b, это подходящее решение.

**Код решения:**

#include **<iostream>**#include **<utility>**#include **<cmath>**#include **"vector"**#include **"cassert"  
  
using namespace** std;  
  
vector<vector<**int**>> getSolution(**int** n, **int** b);  
  
**int** getVSum(vector<**int**> v) {  
 **int** sum = 0;  
 **for** (**auto** i: v) {  
 sum += i;  
 }  
 **return** sum;  
}  
  
  
**void** \_getPlacementsRepeat(**int** k, vector<**int**> &setAvailable,  
 vector<**int**> setChosen,  
 vector<vector<**int**>> &placements) {  
 **if** (setChosen.size() < k) {  
 **for** (**int** j = 0; j < setAvailable.size(); j++) {  
 vector<**int**> setChosenCopy = setChosen;  
  
 setChosenCopy.push\_back(setAvailable[j]);  
  
 \_getPlacementsRepeat(k, setAvailable, setChosenCopy,  
 placements);  
 }  
 } **else** placements.push\_back(setChosen);  
}  
  
**void** getPlacementsRepeat(**int** k, vector<**int**> initialSet,  
 vector<vector<**int**>> &placements) {  
 vector<**int**> setChosen;  
 \_getPlacementsRepeat(k, initialSet, setChosen, placements);  
}  
  
**void** test() {  
 vector<vector<**int**>> testSolution**{**vector<**int**>**{**1, 2**}**,  
 vector<**int**>**{**2, 1**}}**;  
 vector<vector<**int**>> solution = getSolution(2, 3);  
  
 **assert**(testSolution == solution);  
}  
  
**int** main() {  
 test();  
  
 **int** n, b;  
 cout << **"Введите n и b:"** << **'\n'**;  
 cin >> n >> b;  
 vector<vector<**int**>> potentialSolutions = getSolution(n, b);  
  
 **for** (**auto** i: potentialSolutions) {  
 **for** (**auto** j: i)  
 cout << j << **" "**;  
 cout << **'\n'**;  
 }  
}  
  
vector<vector<**int**>> getSolution(**int** n, **int** b) {  
 vector<**int**> appropriateValues(b);  
 **for** (**int** i = 0; i <= b; i++) {  
 appropriateValues[i] = i + 1;  
 }  
  
 vector<vector<**int**>> potentialSolutions;  
 getPlacementsRepeat(n, appropriateValues, potentialSolutions);  
  
 **for** (**int** i = 0; i < potentialSolutions.size(); i++) {  
 **if** (getVSum(potentialSolutions[i]) != b) {  
 potentialSolutions.erase(potentialSolutions.begin() + i);  
 i--;  
 }  
 }  
 **return** potentialSolutions;  
}

**Пример работы программы:**

Введите n и b:

4 7

1 1 1 4

1 1 2 3

1 1 3 2

1 1 4 1

1 2 1 3

1 2 2 2

1 2 3 1

1 3 1 2

1 3 2 1

1 4 1 1

2 1 1 3

2 1 2 2

2 1 3 1

2 2 1 2

2 2 2 1

2 3 1 1

3 1 1 2

3 1 2 1

3 2 1 1

4 1 1 1

**Вывод:** в ходе лабораторной работы мы научились решать задачи с помощью комбинаторных объектов.