МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
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-223
Пахомов Владислав Андреевич

Проверили:
ст. пр. Рязанов Юрий Дмитриевич
ст. пр. Бондаренко Татьяна Владимировна

Белгород 2023 г.

Лабораторная работа №2.2
Задачи выбора
Вариант 10

Цель работы: приобрести практические навыки в использовании алгоритмов порождения комбинаторных объектов при проектировании алгоритмов решения задач выбора.

№1. Ознакомиться с задачей.

Задана точка в узле целочисленной решетке размером $n \times n$. Расположить всеми возможными способами k точек в узлах решетки так, чтобы расстояние от каждой до заданной было одинаковым.

№2. Определить класс комбинаторных объектов, содержащих решение задачи (траекторию задачи).
Сформируем массив points (размер - $n \times n$), который содержит все точки решётки. По условию необходимо взять только k точек. Множество точек фиксированно и их порядок важен (они должны быть упорядочены и не должны повторяться), траектории задачи получим в результате формирования **сочетаний** из множества points по k.

№3. Определить, что в задаче является функционалом и способ его вычисления.

Функционал - массив расстояний от точки в траектории до точки A.
Рассмотрим пример. Пусть $k = 3; A = (3, 4); path = \{(1, 2), (6, 7), (3, 3)\}$
Создадим множество distances.
$distances_i = DISTANCE(A, path_i)$, где DISTANCE - функция вычисления квадрата расстояния между двумя точками.
$distances_0 = (3 - 1)^2 + (4 - 2)^2 = 8$
$distances_1 = (3 - 6)^2 + (4 - 7)^2 = 18$
$distances_2 = (3 - 3)^2 + (4 - 3)^2 = 1$
$distances = \{8, 18, 1\}$ - получили функционал.

№4. Определить способ распознавания решения по значению функционала.

Если все элементы функционала $distances$ равны между собой, значит решение найдено.

№5. Реализовать алгоритм решения задачи.

```
#include "../alg.h"

#define DISTANCE(a, b) \
(pow2(max(b.x, a.x) - min(b.x, a.x)) +\
pow2(max(b.y, a.y) - min(b.y, a.y)))

typedef std::vector<Point> Path;
typedef std::vector<unsigned long long> Functional;
```

```
static unsigned long long max(unsigned long long a, unsigned long long b) {
	return a > b ? a : b;
}

static unsigned long long min(unsigned long long a, unsigned long long b) {
	return a < b ? a : b;
}

static unsigned long long pow2(unsigned long long a) {
	return a * a;
}

// Генерация траекторий задачи
static std::vector<Path> getPaths(size_t n, size_t k) {
	// Массив точек решётки
	std::vector<Point> points;

	// Заполняем его всеми точками решётки.
	for (long long x = 0; x < n; x++)
		for (long long y = 0; y < n; y++)
			points.push_back({x, y});

	// Такие сочетания получить невозможно - возвращаем пустой массив траекторий
	if (points.size() < k || k == 0) return {};

	return getCombinations(points, k);
}

// Генерация функционалов задачи
static std::vector<std::pair<Functional, Path>> getFunctionals(std::vector<Path> paths, Point a) {
	std::vector<std::pair<Functional, Path>> answer;

	// Каждой траектории задачи поставим в соответствие функционал
	// Функционал - массив расстояний от точек в траектории до исходной точки a
	for (auto &path: paths) {
		Functional distances;

		// Определяем расстояние от каждой точки в траектории до точки a
		for (auto &point : path)
			distances.push_back(DISTANCE(a, point));

		// Сохраняем полученный функционал
		answer.push_back({distances, path});
	}

	return answer;
}

// Определяем траектории
size_t getDecision(size_t n, Point a, size_t k) {
	// Генерация траекторий
	auto paths = getPaths(n, k);

	// Генерация функционала
	auto functionals = getFunctionals(paths, a);

	size_t decisions = 0;
	// Если все расстояния в функционале одинаковы, решение найдено
	for (auto &pair : functionals) {
		auto distances = pair.first;
		bool areEqual = true;

		for (auto &dist : distances)
			areEqual &= dist == distances[0];

		decisions += areEqual;
	}
```

```
        return decisions;
}
```

№6. Подготовить тестовые данные и решить задачу.

```
#include "../../libs/alg/alg.h"

int main() {
        assert(getDecision(1, (Point){0, 0}, 1) == 1);
        assert(getDecision(1, (Point){0, 0}, 12) == 0);
        assert(getDecision(5, (Point){3, 2}, 1) == 25);
        assert(getDecision(5, (Point){3, 3}, 2) == 27);
        assert(getDecision(5, (Point){3, 3}, 4) == 4);
        assert(getDecision(0, (Point){0, 0}, 0) == 0);
}
```

```
/Users/vlad/Desktop/C/discrete_math/cmake-build-debug/bin/lab6_task6

Process finished with exit code 0
```

Вывод: в ходе лабораторной работы приобрели практические навыки в использовании алгоритмов порождения комбинаторных объектов при проектировании алгоритмов решения задач выбора.