Федеральное государственное автономное образовательное учреждение

высшего образования

«Нижегородский государственный университет

им. Н.И. Лобачевского»

Институт информационно-вычислительных технологий математики и механики.

Отчёт по лабораторной работе

Решение задачи «Просачивания» (Percolation)

Выполнил:

студент ф-та ИИТММ гр. 8203-2

Кипаренко И.В

Проверил:

ассистент каф. МО ЭВМ, ИИТММ

Сиднев А.А.

Нижний Новгород

2016 г.

**Содержание**

[Введение](#h.gjdgxs)

[Постановка задачи](#h.30j0zll)

[Руководство пользователя](#h.1fob9te)

[Руководство программиста](#h.3znysh7)

[Описание структур данных](#h.3znysh7)

[Описание алгоритмов](#h.2et92p0)

Описание структуры программы

[Заключение](#h.3dy6vkm)

Литература

Приложения

Приложение 1

[Приложение 2](#h.17dp8vu)

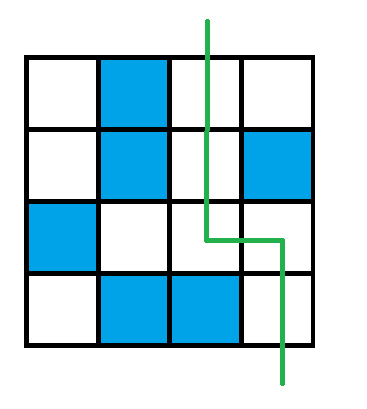
**Введение**

В этой лабораторной работе, я научился работать с системами контроля версий, разрабатывать библиотеки, и проводить вычислительные эксперименты.

Значение моей лабораторной работы для всего мира, неоспоримо очень велико, так как она продолжает традицию преемственности поколений студентов.

**Постановка задачи**

Есть поле , состоящее из ячеек, каждая может быть в открытом или закрытом состоянии. Скажем, что из верхней части поля, есть путь в нижнюю, если можно проходя только по открытым клеткам, и не выходя за пределы поля, попасть с какой-то клетки, верхней строки, в какую-то клетку нижней строки.



Пример. На картинке выше, поле размера , закрытые клетки отмечены синим, и путь зеленой линией.

Задача:

1. Реализовать программу, проверяющую, есть ли путь в данном поле.
2. Использовать её, чтобы поставить вычислительный эксперимент, к чему стремится отношение открытых ячеек к общему числу ячеек, если все ячейки изначально закрыты, мы открываем случайную из закрытых пока не найдём путь.

**Руководство пользователя**

Переменная EXP\_SIZE определяет количество экспериментов.

Переменные n , m , задают соответственно размеры поля.

В результате программа печатает на экран, среднее число, процентов открытых клеток, при котором был найден путь, а также время вычисления.

**Руководство программиста**

***Описание структуры программы***

Программа состоит из :

1. Библиотеки, с реализацией класса разделенных множеств (SepSet).
2. Проекта с тестами на основе библиотеки gtest.
3. Проекта, реализующего вычислительный эксперимент (X\_Percolation\_Sample)

***Описание структур данных***

Класс SepSet, реализует разделенные множества, на основе древесной сбалансированной структуры, реализованной на массиве.

У него есть методы :

* add() , который создаёт, новое одноэлементное множество (singleton)
* merge() , который объединяет два множества в одно.
* whereI() , который возвращает, в каком множестве находится данный элемент.

***Описание алгоритмов***

В программе был реализован алгоритм просачивания (percolation) с применением разделенных множеств.

Алгоритм :

1. Добавляем фиктивную строчку, выше самой верхней, и ниже самой нижней.
2. Добавляем номера всех элементов этих строчек, в два множества (верхнее и нижнее)
3. Для каждой открытой ячейки, создаём одноэлементное множество, и если есть соседние открытые, то объединяем их множества.
4. Если в конце, верхнее и нижнее множество, объединились, значит есть путь.

**Заключение**

Была реализована программа, и проведены следующие вычислительные эксперименты.

* Поле , 1.5 миллиона экспериментов, средний процент 0.59257
* Поле , 0.8 миллиона экспериментов, средний процент 0.59278

**Литература**

1. Столлингс, В. Структурная организация и архитектура компьютерных систем, 5-е изд.: Пер. с англ. — М.: Издательский дом «Вильямс», 2002. — 896 с.: ил. — Парал. тит. англ.
2. Johnson M. Superscalar Microprocessor Design. — Englewood Cliff, New Jersey: Prentice Hall, 1991.
3. Касперски К. Техника оптимизации программ. Эффективное использование памяти. — СПб.: БХВ-Петербург, 2003. — 464 с.: ил.
4. Stone H. High performance Computer Architecture. — Reading, MA: Addison-Wesley, 1993.
5. Tullsen D.M., Eggers S.J. Effective Cache Prefetching on a Bus-Based Multiprocessor. — ACM Transactions on Computer Systems, pp. 57-88, Feb 1995.
6. Chandra D., Guo F., Kim S., Solihin Y. Predicting inter-thread cache contention on a chip multi-processor architecture. — Proceedings of the 11th International Symposium on High Performance Computer Architecture (HPCA), pp. 340–351, Feb 2005.
7. Press W., Teukolsky S., Vetterling W., Flannery B. Numerical Recipes in C. The Art of Scientific Computing. Second Edition. — Cambridge University Press, 1992.
8. Камаев А.М., Сиднев А.А., Сысоев А.В. Об одном подходе к анализу эффективности приложений // Труды 50-й научной конференции МФТИ «Современные проблемы фундаментальных и прикладных наук»: Часть I. Радиотехника и кибернетика. - М.: МФТИ, 2007.
9. Debugging and performance monitoring. Intel® 64 and IA-32 Architectures Software Developer’s Manual. Volume 3B: System Programming Guide, Part 2. May 2007. — [http://www.intel.com/products/processor/manuals/]
10. Юнаковский А.Д. Начала вычислительных методов для физиков. – Н. Новгород: ИПФ РАН, 2007.

**Приложения**

***Приложение 1***

***Приложение 2***