



HÁSKÓLINN Í REYKJAVÍK
REYKJAVIK UNIVERSITY

FALL 2020

T-301-REIR, REIKNIRIT

S1: PERCOLATION

(KETAMINE ADDICT), (ILLUGIF@RU.IS)
(APPLE FANBOY), (PETURS19@RU.IS)

STUDY GROUP: ???

SEPTEMBER 10, 2021

TA'S: ALL OF THEM

1 Introduction

The objective of the exercise is to find out if a N-by-N grid percolates, or better said, find the "magic" ratio between open and closed sites that makes the system percolate. We performed the exercise to give us a better understanding of the problem at hand and understand how to utilize the different union-like structures. Our aim was to code the problem with different union like structures and then test it vigorously by using our Stats class.

Percolation

Percolation (Útskýra hvað í fokbanum percolation er - copya frá lýsingu ?)

1.1 Setup and Methods

We did the project on "macc details" using IntelliJ from JetBrains for Java development and we used Java 11. We started on figuring out how we want to store all of the data and how we can do that efficiently! We started with an integer grid so we could get the row and col and get the value but a matrix of integers takes a lot of memory, so we settled on using a boolean matrix to keep track whether the site is open or not, but that created a problem, how are we going to find the index for the union data structure? Simple! We devised a plan! Since we know that it's an N-by-N grid then we can calculate the index for the data structure, the formula is as such:

$$index = (row * N) + col \tag{1}$$

After we found that out we started programming, we then figured out that we need to find an efficient way to store pairs of row number and column number, we created a private Pair class within the Percolation class, The pair class is as simple as it gets, it stores the row number and the column number.

When we finished that we went about writing the open function, the open function simply opens the site and then checks if any other neighbouring cells are open, if so we connect them to the union data structure. But how are we going to find the neighbours? By writing methods! Private methods to be exact, We wrote a method that gets the neighbouring cells row number and column number and then we wrote another private method that checks if the row numbers and column numbers are valid within the N-by-N matrix the system is in.

Private methods

1.2 Implementation

2 Results

Percolation constant

2.1 Quick-Find

Results

Time complexity Running time as a function of N and T : \sim

Table 1: Using Java11 LTS

(a) QuickFind results

N	T	Total Running Time(s)	Confid.(low)	Avg Threshold	Confid.(high)
200	100	20.886	0.5916614856943488	0.593601000905037	0.5955405161157251
200	200	58.478	0.5915249435319724	0.5928471261262893	0.5941693087206062
400	100	301.269	0.5922470075078308	0.5933678120374679	0.594488616567105
400	200	899.393	0.5915991882347434	0.5923998117446899	0.5932004352546364

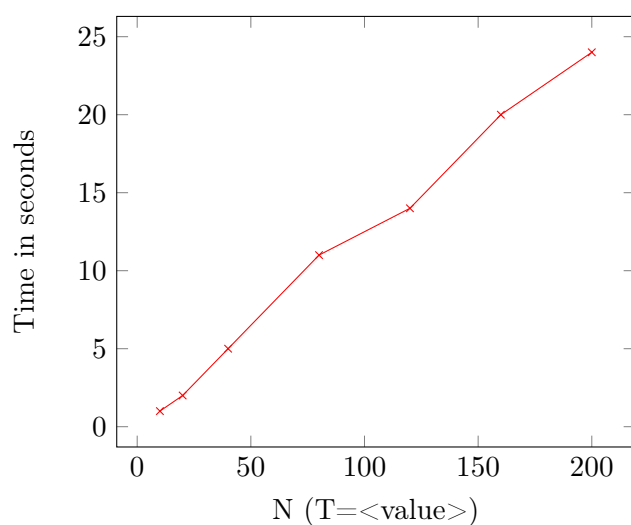


Figure 1: This is the caption for the plot

A chart should be included here.

Figure 2: This is the caption for the chart.

2.2 Weighted Quick-Union

2.3 Memory Usage

2.4 Discussion/Conclusions

3 About This Solution

Have you taken (part of) this course before:

Hours to complete assignment (optional):

3.1 Quiz on Collaboration

1. How much help can you give a fellow student taking REIR?

(a) None. Only the TAs can help.

(b) You can discuss ideas and concepts but students can get help debugging their code only from a TA, or student who has already passed REIR.

Table 2: Using Java11 LTS

(a) Weighted QuickUnion results

N	T	Total Running Time(s)	Confid.(low)	Avg Threshold	Confid.(high)
200	100	0.373	0.5895962947979569	0.591609999537468	0.593623704276979
200	200	0.659	0.5912201780959143	0.5912201780959143	0.594014821560763
400	100	1.415	0.5906189738214016	0.5918477481603622	0.5930765224993229
400	200	2.302	0.592641312851526	0.5934378761053085	0.594234439359091

(c) You can help a student by discussing ideas, selecting data structures, and debugging their code.

(d) You can help a student by emailing him/her your code.

Answer:

2. What is the expectation when partnering?

(a) You and your partner split the assignment between you and solve the parts separately.

(b) You and your partner discuss all the problems together, but code separately.

(c) You and your partner discuss the problems and write all the code together.

Answer:

3.2 Known Bugs / Limitations.

3.3 Help Received

3.4 Problem Encountered

3.5 Comments