REPORT FOR ALGORITHM & DATA STRUCTURES ELEVATOR PROJECT

GROUP 14: IBRAHIM OYEKAN, SIDDHANT KALASH, JONATHAN CHISMAR.

The source code consists of seven classes in total, namely:

Elevator:

The elevator class handles the primary functionality of the elevator. It starts by sorting external calls with respect to time and destination. It queues the external calls, picks people (of type Person) with corresponding directions at their respective floors, and then executes the internal sort method which sorts people according to proximity of destination. While traveling, the elevator class checks if external calls appear within the time period at which the foremost-priority passenger is being dropped off, and queues these external calls accordingly.

Based on this algorithm, the elevator executes pick-ups and drops people off at their destinations, stopping between journeys to make correlating pick-ups and save time.

Building:

The Building class creates an array of n Floors (of type Floor), and then creates array lists of people appearing at any given time. This information is used by the elevator class to execute movement accordingly.

Floor:

The Floor class creates a Floor data structure according to number of floors required. It contains an isPresent field which tracks what floor the elevator is currently on.

Person:

The Person class creates a person at a specified floor (Floor waiting), assigns to them a destination (Floor goingTo) and a direction accordingly, using information received from the text file. These people also have times assigned to them from each line of the TestCase text file.

REPORT FOR ALGORITHM & DATA STRUCTURES ELEVATOR PROJECT

GROUP 14: IBRAHIM OYEKAN, SIDDHANT KALASH, JONATHAN CHISMAR.

BasicElevator:

The BasicElevator class implements a basic elevator which only moves from the top of the building to the bottom of the building, picks people waiting for the elevator at any given point in time, and drops people off when their destination is reached.

TestCase:

The TestCase class generates a text file with the first line representing n floors in the building and each subsequent line representing a person. The first column represents a time, the second column represent what floor the person is at, and the third column represents what floor the person is going to.

Run Instructions: You will be prompted by the program to enter either the Basic Elevator or the Optimized Elevator, or to quit to the program. Test Cases used are compressed in the zip file, in addition to the TestCase generator program.

Compiler instructions: java -cp "(your directory)/Elevator/StdDraw.jar" (your directory)/Elevator/src/*.java

Run Driver class as it contains the main method.

Data analysis:

The table below shows statistics for the number of movements executed using the basic elevator and the optimized elevator for individual buildings of 5 to 15 floors. Both elevators utilize the same random samplings of people generated from the TestCase class. Each size of building is generated 3 times (Day 1-3) for accuracy, and each TestCase contains 50 people generated at random times and assigned random attributes.

REPORT FOR ALGORITHM & DATA STRUCTURES ELEVATOR PROJECT

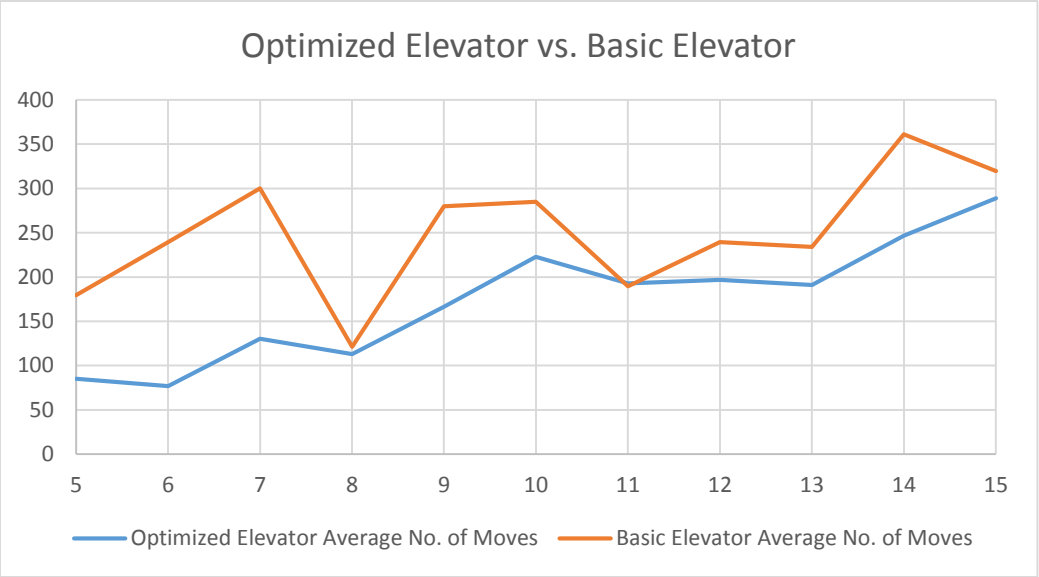
GROUP 14: IBRAHIM OYEKAN, SIDDHANT KALASH, JONATHAN CHISMAR.

Basic Elevator					
		Day 1	Day 2	Day 3	
	No. of Floors	No. of Movements	No. of Movements	No. of Movements	Basic Elevator Average
	5	228	62	248	179
	6	310	86	322	239
	7	416	95	389	300
	8	119	132	113	121
	9	547	118	175	280
	10	182	467	206	285
	11	253	81	235	190
	12	236	167	315	239
	13	119	291	292	234
	14	414	299	371	361
	15	435	349	175	320
Optimized Elevator					
		Day 1	Day 2	Day 3	
	No. of Floors	No. of Movements	No. of Movements	No. of Movements	Optimized Elevator Average
	5	87	82	86	85
	6	103	60	67	77
	7	127	121	143	130
	8	91	113	135	113
	9	197	167	135	166
	10	249	183	236	223
	11	205	111	262	193
	12	231	177	182	197
	13	153	223	197	191
	14	248	262	230	247
	15	291	320	256	289

No. of Floors	Optimized Elevator Average No. of Moves	Basic Elevator Average No. of Moves
5	85	179
6	77	239
7	130	300
8	113	121
9	166	280
10	223	285
11	193	190
12	197	239
13	191	234
14	247	361
15	289	320

REPORT FOR ALGORITHM & DATA STRUCTURES ELEVATOR PROJECT

GROUP 14: IBRAHIM OYEKAN, SIDDHANT KALASH, JONATHAN CHISMAR.



Results show our Optimized Elevator performs much better than the Basic Elevator by delivering the same distribution of people to their destinations while executing a smaller number of movements.

The average number of movements for each building using the optimized elevator is 174, while the average number of movements for each building using the basic elevator is 250. Based on this statistic, our optimized elevator is shown to execute the same amount of work done by the basic elevator under just 70% of the time taken by it, saving both power and time in the process.