LAB Assignment 2
(CS205)
Topic: Recursion and Performance of Algorithm
13th August 2019
Home Assignment: a

a) We have to compare two lift controlling algorithms in terms of number trips required to carry people of a 10 floor building for X hours and average waiting time. It is given that mean rate of arrival of people in each floor varies across floors. Mean rate of arrival in ground floor is 200 per hour and for all other floors this rate is in the range of 20 – 25 per hour (you may generate a random number in the range of 20-25). Arrival of people follows a normal distribution. There are total 1000 users who have office in this building. Every user has a unique id starting from 000 to 999. Office of a particular employee is in xth floor if id%10 == x. A person who is having office in xth (except ground) floor use lift 95% of time for going to either ground floor or xth floor and for 5 % of time for going to any random floor. Lift usage by persons who use to sit in ground floor is on an average 1/20 times compared to other persons. Lift maintains two request queues (inside queue which maintains requests from insiders – maintains destination floor number, time of request and outside queue which maintains request from outsiders –maintains request originated floor number, requested moving direction, time of request). We also assume that as soon as request received from an insider/outsider queues are updated instantaneously. Lift also maintains current state (idle, moving down, moving up). To decide moving direction and next destination it follows following logic. When lift is idle and request received from any floor, lift will set its direction based on if the requested floor is higher/lower compared to where the lift is currently parked and next destination will be set to the floor from which request came from and starts moving. For example lift was parked at 4th floor and a request came from 1st floor to move downward. Outer Queue will have one entry (1, downward, t1). This request will be dequeued from queue and put into service stack. As 4 is > 1, it needs to move downward and next destination is set to 1 and it starts moving downward. However, assume when the lift is near third floor, another request is received from 2nd floor to move downward. Currently outer queue will have an entry (2, downward, t2). Lift controller checks request in outer queue with request in service stack and current floor and found that current moving direction and requested moving direction is same and requested floor is nearer compared to current destination (request originating floor (that is 2nd floor) is closer to 3rd floor compared to the current destination 1st floor) so it dequeue the current request and put on the top of service stack and modify the current destination as 2nd. When current destination is served, lift controller pops that from stack top. So we can push request to service stack only when that service can be served in the path of current destination. When service stack becomes empty it will check for any entry either in outer queue or in insider queue and controller will place that entry into service stack which was requested earlier. In case both queue have same time, inside queue will be given more priority. Lift can carry at most 10 people at a time. You can assume that lift takes 20 seconds to move up/down one floor. Maintain an array per floor which will describe which employee is current in which floor. Assume at any point 4th floor array looks like 004 234 454 654 235. It means the following employees (004 234 454 654 235) are currently in 4th floor. The Box–Muller method uses two independent random numbers *U* and *V* distributed uniformly on (0,1). Then the two random variables *X* and *Y*

$$X = \sqrt{-2\ln U}\,\cos(2\pi V), \qquad Y = \sqrt{-2\ln U}\,\sin(2\pi V).$$

will both have the standard normal distribution, and will be independent. Normal distribution can be generated as *X* = *μ* + *σZ*, For ground floor *μ=200, σ=20 and for other floor μ=(20-25), σ=4. Assume that request can be generated at any integral second. Also assume there is a overhead of 10 seconds for opening and closing lift door.*

Algorithm1: After lift door opens, it remains open for a fixed amount of time say 1 min if capacity is not full. However if the lift becomes full before one minute then door gets closed immediately.

Algorithm2: After lift door opens, it finds time to arrive 10 persons as per arrival distribution and set initial waiting time as min(2 min, time for arrival of 10 persons) [It will not check actual time of arrival, rather it will estimate that using the knowledge of arrival distribution]. At every 10 seconds, it will adjust waiting time by reducing 10 seconds. However, if k persons arrived in last 10 seconds then additional 10k seconds will be reduced from waiting time. When waiting time reaches 0 or capacity of lift becomes full lift doors closed.

Expected format of output:

	Algorithm1	Algorithm2
1 hour	(Number of trip, average waiting time)	(Number of trip, average waiting time)
2 hour	(Number of trip, average waiting time)	(Number of trip, average waiting time)
3 hour	(Number of trip, average waiting time)	(Number of trip, average waiting time)
Up to 10 hours	(Number of trip, average waiting time)	(Number of trip, average waiting time)

Upload : Ass2a.c