

Housing Queue Simulator

University of North Texas provides limited off-campus apartments to undergraduate students. The housing system uses a waiting list for apartment assignments. According to the policy, when a possible match between applicant and apartment is found, the applicant is contacted via email and must decide whether to accept the apartment. If the applicant does not accept, he or she is placed back at the beginning of the waiting list and must wait again.

(a) We will simulate this process using a queue and simplify the rules. We will ignore the different types of apartments, and simply give each apartment a quality rating. Moreover, we will ignore the realistic constraints of the applicants, and simply have each applicant store a quality threshold. We will also assume that each apartment, once accepted, will be occupied for a random number of school years between 1 and 4, and that vacant apartments are filled between school years. This means our simulation will use the school year as its unit of time.

- Assume k students apply for housing each school year. Each student node should contain a randomly generated quality threshold (a random double between 0 and 1), and an ID number (or String) to keep track. Get N and k from the user via command line arguments.
- Create a list of N apartments. Each apartment should store its ID number or String and two pieces of information: its quality score (a random double between 0 and 1), and how many school years left in its occupancy (0 means it is vacant and ready to be assigned).
- At the end of each school year, decrement the number of school years remaining for each occupied apartment, and run the assignment process for each vacancy.
- The assignment process should dequeue a student, check if the student “accepts” the apartment (is the apartment’s quality value above the student’s threshold?). If the student accepts, set the apartment to occupied for a random period of between 1 and 4 school years and delete the student node. If the student rejects the apartment, enqueue the student, dequeue the next student and repeat.
- If all students reject an apartment, move on to the next apartment. This means there may be vacancies in terrible apartments even if there are still students on the waiting list.
- The assignment process repeats until either there are no vacancies, all students are housed, or all unhoused students have seen all apartments. Once either of these occur, move to the next school year.
- To make the simulation more realistic (and to avoid an infinite loop during the assignment step), each time a student is put back on the waiting list, multiply his or her threshold by 0.9. This will simulate the student growing desperation.
- Output statistics after each school year. Display the number of apartments vacated that year, the number of apartments newly filled. When k is small, output the steps of the process as they occur.

(b) Add a feature to the code that outputs the average years spent on the waiting list. Experiment with N and k to see how the two parameters affect the waiting time.

You must write your own `MyQueue` class for this problem. Do not use the built-in `Queue` interface.