Gavin Kyte
CSCI 220
Store Simulation Analysis + Big O Summary

In response to the owner's question of how to best layout their store's registers, the face-value best solution is to keep the eight normal registers, but this comes with caveats.

For the given scenarios, the simulation results show that the average wait time is lowest with eight normal registers and highest with four normal and eight self checkout; the difference is roughly seventeen seconds. Further down this road, a radical scenario of entirely self-checkout lanes results in horrible average wait times of over three and a half minutes.

That said, it may still be worth trading for four self-checkout lanes if there exists a subjective factor that is more important than a four second increase in wait time, such as maximum line length. In the default scenario, this is three, but the six and four combo has comparable wait times and maxes out at only two. Is that significant enough to warrant change? Probably not, considering that only one register had that max of three and there probably isn't considerable enough foot traffic to create a scenario where the registers were overloaded. It's worth noting that in the eight, zero combo, 114 customers complained; comparatively, the six, four combo resulted in 153 complaints.

In conclusion, I suggest doing nothing, OR adding three normal registers because this results in 0 complaints. What do you prioritize? 0 complaints, or realistic wait times? Depends on your resources.

Gavin Kyte
CSCI 220
Store Simulation Analysis + Big O Summary

PriorityQueue

- Enqueue()
  - Best: O(1)
  - Worst: O(n)
  - Average: O(n) * Usually n/2 for unordered data
- Dequeue()
  - Best: O(1)
  - Worst: O(1)
  - Average: O(1)
- Delete()
  - Best: O(n)
  - Worst: O(n)
  - Average: O(n)

RegisterQueue

- Enqueue()
  - Best: O(1)
  - Worst: O(1)
  - Average: O(1)
- Dequeue()
  - Best: O(1)
  - Worst: O(1)
  - Average: O(1)
- SeeNext()
  - Best: O(1)
  - Worst: O(1)
  - Average: O(1)
- Delete()
  - Best: O(n)
  - Worst: O(n)
  - Average: O(n)