人気ラーメン店にできる行列

1116170036　長江優輝

1.　課題内容

　この課題は，人気のあるラーメン店の行列をシミュレーションすることである．シミュレーションの条件は以下の3つである．

1.　席はカウンターのみで，8つとする．

2.　客はランダムに訪れると仮定する．ただし，訪れるのは12時から13時の間のみとする．そして，10分あたりの客は3人～10人のいずれかとする．

3.　1人が席についてからの滞在時間は，課題により異なるものとする．

2.　課題1：客の滞在時間が一定（1人あたり17分）の場合

2.1　 解法

　本課題はJavaFXを利用してプログラムを作成した．グラフ表示もJavaFXで行っている．図1に客の人数と，その来訪時間をランダムで決めるソースコードを示す．図2に行列のシミュレーションをするソースコードを示す．

図1では，10分あたりに訪れる客の人数をrandom関数で決定している．人数が分かったので，for文をループして人数分の来訪時間を，random関数を用いて決定した数値を，配列visitors[]に書き込んでいく．最後に，バブルソートを用いて来訪時間を早い順に並べなおすという作業を行っている．

図2では，for文を2時間分，つまり7200回ループさせている．また，重要なものとして，8行7200列の多次元配列congestion[][]というものがある．これは，席ごとに，ある時間の時点にその席が空いているかどうかを識別するための配列である．1ループごとにcongestion[][]に客が座っていない場合は0を書き込んでいく．変数timeの数値とvisitors[]の数値が一致すると，for文を用いてカウンターを0番目から調べ始めて，空席を探す．空いている席があれば，そのcongestion[][]に客が帰る時間まで座っている状態として，1を書き込む．空席がない場合は変数lineを1増加させる．この作業を7200回繰り返す．

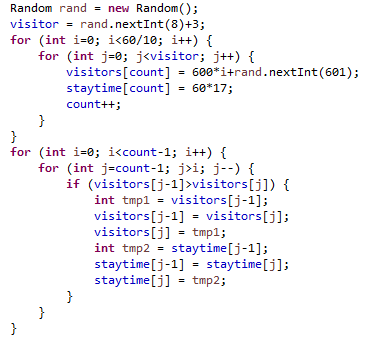


図1　客の人数と，その来訪時間を決めるプログラム部分

図2　行列のシミュレーションをするプログラム部分

2.2　結果

　図3に行列の長さの時間的変化のグラフを示す．これは客の人数が48人であった場合のものである．行列の長さは最大で20人であり，13：30頃には行列は無くなっている．

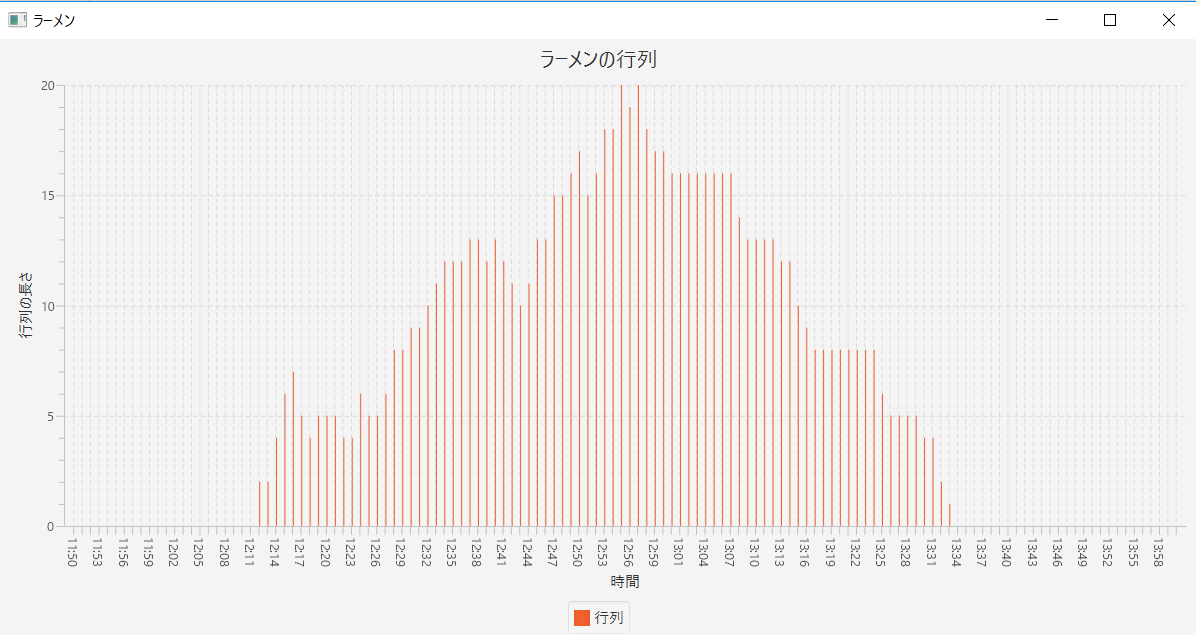


図3　行列の長さの時間的変化

2.3　考察

　この課題では客の滞在時間が17分と決まっているため，2時間の間では，1席につきおよそ7人座ることができると分かる．そのため，8席あっても2時間で回すことができるのは56人程度である．つまり，もし客が来る人数として最大である，60人が来た場合は，14:00時点でも4人前後は行列が残ってしまう．

3.　課題2：客の滞在時間が異なる（1人あたり10～24分）の場合

3.1　解法

　プログラムの大半は，課題1と同じである．図4に変更した部分のソースコードを示す．変更点としては，客の滞在時間を，ランダム関数を用いて決定しているだけである．

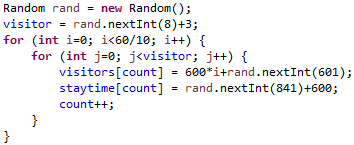


図4　課題1から変更したプログラム部分

3.2　結果

　図5に行列の長さの時間的変化のグラフを示す．これは客の人数が60人であった場合のものである．行列の長さは最大で31人であり，14:00の時点でも，まだ列が2人残っている．

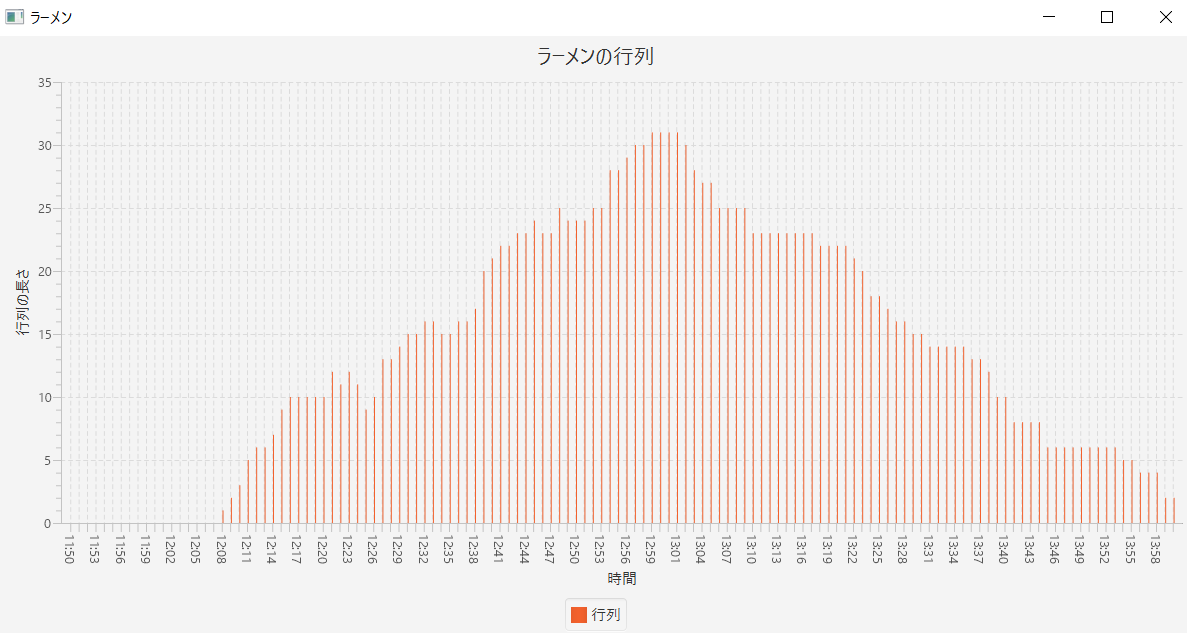


図5　行列の長さの時間的変化

3.3　考察

　客が最大の60人来たために2時間では全員がラーメンを食べることはできなかった．しかし，客の滞在時間が全体的に短ければ，客の人数が60人でも14:00までには行列が無くなることもあるようである．

4.　発展課題：列が長くなる場合に席を増やす．

4.1　解法

　行列が5人を超えた場合に席を2つ増やすようにプログラムを変更した．一旦席が増えると最後まで席が減ることは無い．図6に追加プログラムを示す．これを図2の下から9行上の部分に追加している．if文を用いて，行列が5人を超えたとき，席を2つ追加したfor文がループされるようにした．

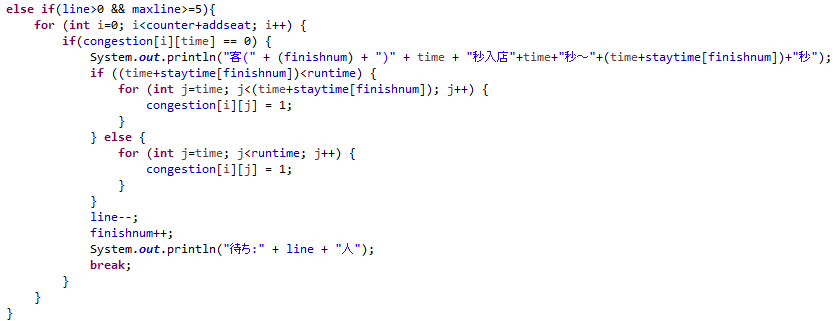


図6　課題2から追加したプログラム部分

4.2　結果

　図6に行列の長さの時間的変化のグラフを示す．これは客の人数が60人であった場合のものである．行列の長さは最大で26人であり，13:40頃には行列は無くなっている．

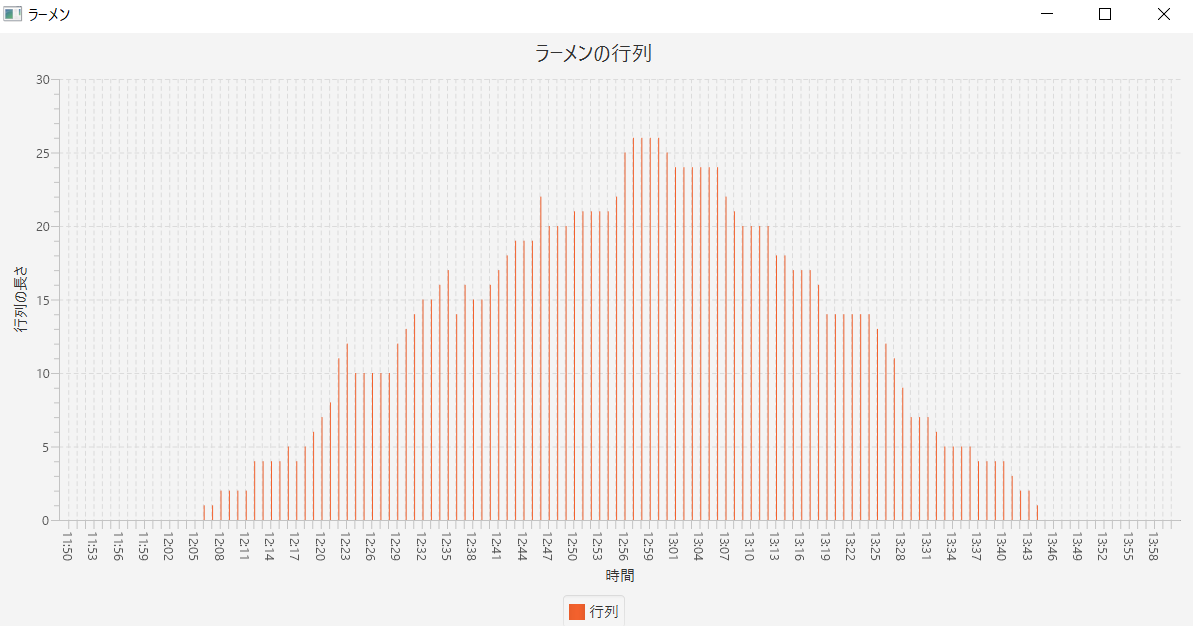


図6　行列の長さの時間的変化

4.3　考察

　客が最大人数の60人の場合にも14：00までには必ず行列は無くなるようになった．2席追加されるだけでも混雑の解消には大きく効果があるようである．

5.　参考文献

Javaで指定範囲の乱数を生成する：Random.nextInt() | UX MILK：https://uxmilk.jp/47989

Javaでバブルソートのプログラムを作成する方法【初心者向け】

https://techacademy.jp/magazine/19444