

Gebze Technical University
Computer Engineering

CSE 222
2017 Spring

HOMEWORK 04 REPORT

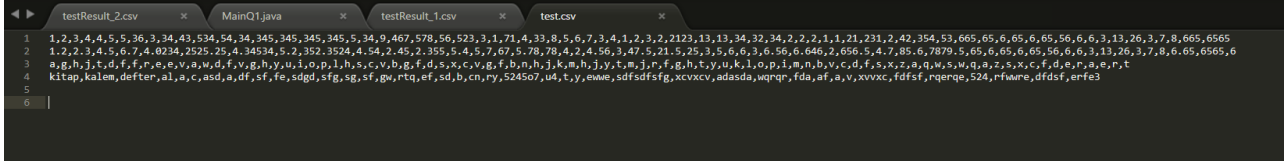
Elif Şeyma ARMAĞAN
151044042

Course Assistant:
Nur Banu ALBAYRAK

Q1)

1. Test Cases

test.csv dosyası:



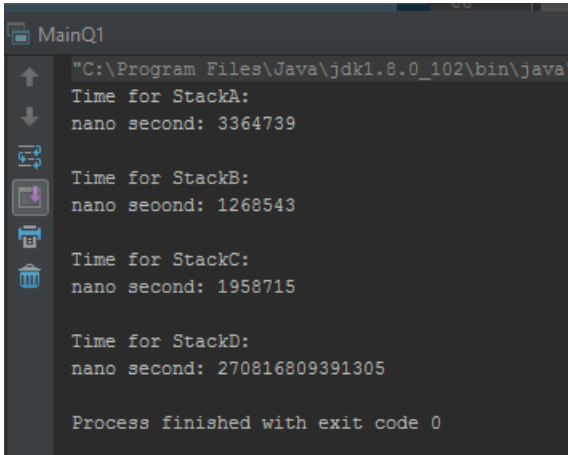
```
1 1,2,3,4,4,5,5,36,3,34,43,534,54,34,345,345,345,345,5,34,9,467,578,56,523,3,1,71,4,33,8,5,6,7,3,4,1,2,3,2,2123,13,13,34,32,34,2,2,2,1,1,21,231,2,42,354,53,665,65,6,65,6,65,56,6,6,3,13,26,3,7,8,665,6565
2 1,2,2,3,4,5,6,7,4,8234,2525,25,4,34534,5,2,352,3524,4,54,2,45,2,355,5,4,5,7,67,5,78,78,4,2,4,56,3,47,5,21,5,25,3,5,6,6,3,6,56,6,646,2,656,5,4,7,85,6,7879,5,65,6,65,6,65,56,6,6,3,13,26,3,7,8,6,65,6565,6
3 a,g,h,j,t,d,f,f,r,e,e,v,a,w,d,f,v,g,h,y,u,i,o,p,l,h,s,c,v,b,g,f,d,s,x,c,v,g,f,b,n,h,j,k,m,h,j,y,t,m,j,r,f,g,h,t,y,u,k,l,o,p,i,m,n,b,v,c,d,f,s,x,z,a,q,w,s,w,q,a,z,s,x,c,f,d,e,r,a,e,r,t
4 kitap,kalem,defter,al,a,c,asd,a,df,sf,fe,sdgd,sfg,sg,sf,gw,rtq,ef,sd,b,cn,ry,5245o7,u4,t,y,ewms,sdfsdfsg,xvcxvcy,adasda,wqrqr,fda,af,a,v,xvxc,fdfs,f,qqrqe,524,rflwre,dfdsf,enfe3
5
6
```

Bütün stack’ler bu veriler ile test edilmiştir.

2. Running and Results

Ödev pdf’inde belirtildiği şekilde test edince zamanlar şöyle çıkıyor:

Run 1:



```
MainQ1
"C:\Program Files\Java\jdk1.8.0_102\bin\java"
Time for StackA:
nano second: 3364739

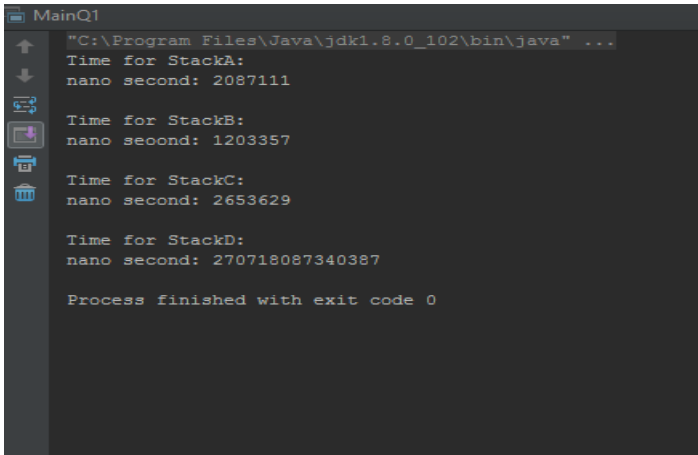
Time for StackB:
nano second: 1268543

Time for StackC:
nano second: 1958715

Time for StackD:
nano second: 270816809391305

Process finished with exit code 0
```

Run 2:



```
MainQ1
"C:\Program Files\Java\jdk1.8.0_102\bin\java" ...
Time for StackA:
nano second: 2087111

Time for StackB:
nano second: 1203357

Time for StackC:
nano second: 2653629

Time for StackD:
nano second: 270718087340387

Process finished with exit code 0
```

testResult_1.csv dosyası ise şu şekilde:

```
4 42,erfe3,didsf,rfiame,524,qnerqe,fdfs,f,xxxc,v,a,af,fda,wqqr,adasda,xcvxcv,sdfsdfsf,ewwe,y,t,u4,524507,ry,cn,b,sd,ef,rtq,gw,sf,sg,sfg,sdgd,fe,sf,df,a,asd,c,a,al,defter,kalem,kitap
5 74,6565,665,8,7,3,26,13,3,6,6,56,65,6,65,665,53,354,42,2,231,21,1,1,2,2,2,34,32,34,13,13,2123,2,3,2,1,4,3,7,6,5,8,33,4,71,1,3,523,56,578,467,9,34,5,345,345,345,34,54,534,43,34,3,36,5,5,4,4,3,2,1
6 43,56,0,65,0,6,0,65,0,6,0,65,0,7879,5,85,6,4,7,656,5,2,0,6,646,6,56,3,0,6,0,6,0,5,0,3,0,25,0,21,5,47,5,3,0,4,56,2,0,4,0,78,0,5,78,67,0,7,0,5,0,5,4,2,355,2,45,4,54,352,3524,5,2,4,34534,2525,25,4,0234,6,7,4,5,2,3,1,2
7 92,t,r,e,a,r,e,d,f,c,x,s,z,a,q,w,s,w,q,a,z,x,s,f,d,c,v,b,n,m,l,p,o,l,k,u,y,t,h,g,f,r,j,m,t,y,j,h,m,k,j,h,n,b,f,g,v,c,x,s,d,f,g,b,v,c,s,h,l,p,o,l,u,y,h,g,v,f,d,w,a,v,e,e,r,f,f,d,t,j,h,g,a
8 42,erfe3,didsf,rfiame,524,qnerqe,fdfs,f,xxxc,v,a,af,fda,wqqr,adasda,xcvxcv,sdfsdfsf,ewwe,y,t,u4,524507,ry,cn,b,sd,ef,rtq,gw,sf,sg,sfg,sdgd,fe,sf,df,a,asd,c,a,al,defter,kalem,kitap
9 74,6565,665,8,7,3,26,13,3,6,6,56,65,6,65,665,53,354,42,2,231,21,1,1,2,2,2,34,32,34,13,13,2123,2,3,2,1,4,3,7,6,5,8,33,4,71,1,3,523,56,578,467,9,34,5,345,345,345,34,54,534,43,34,3,36,5,5,4,4,3,2,1
10 43,56,0,65,0,6,0,65,0,6,0,65,0,7879,5,85,6,4,7,656,5,2,0,6,646,6,56,3,0,6,0,6,0,5,0,3,0,25,0,21,5,47,5,3,0,4,56,2,0,4,0,78,0,5,78,67,0,7,0,5,0,5,4,2,355,2,45,4,54,352,3524,5,2,4,34534,2525,25,4,0234,6,7,4,5,2,3,1,2
11 92,t,r,e,a,r,e,d,f,c,x,s,z,a,q,w,s,w,q,a,z,x,s,f,d,c,v,b,n,m,l,p,o,l,k,u,y,t,h,g,f,r,j,m,t,y,j,h,m,k,j,h,n,b,f,g,v,c,x,s,d,f,g,b,v,c,s,h,l,p,o,l,u,y,h,g,v,f,d,w,a,v,e,e,r,f,f,d,t,j,h,g,a
12 42,erfe3,didsf,rfiame,524,qnerqe,fdfs,f,xxxc,v,a,af,fda,wqqr,adasda,xcvxcv,sdfsdfsf,ewwe,y,t,u4,524507,ry,cn,b,sd,ef,rtq,gw,sf,sg,sfg,sdgd,fe,sf,df,a,asd,c,a,al,defter,kalem,kitap
13 74,6565,665,8,7,3,26,13,3,6,6,56,65,6,65,665,53,354,42,2,231,21,1,1,2,2,2,34,32,34,13,13,2123,2,3,2,1,4,3,7,6,5,8,33,4,71,1,3,523,56,578,467,9,34,5,345,345,345,34,54,534,43,34,3,36,5,5,4,4,3,2,1
14 43,56,0,65,0,6,0,65,0,6,0,65,0,7879,5,85,6,4,7,656,5,2,0,6,646,6,56,3,0,6,0,6,0,5,0,3,0,25,0,21,5,47,5,3,0,4,56,2,0,4,0,78,0,5,78,67,0,7,0,5,0,5,4,2,355,2,45,4,54,352,3524,5,2,4,34534,2525,25,4,0234,6,7,4,5,2,3,1,2
15 92,t,r,e,a,r,e,d,f,c,x,s,z,a,q,w,s,w,q,a,z,x,s,f,d,c,v,b,n,m,l,p,o,l,k,u,y,t,h,g,f,r,j,m,t,y,j,h,m,k,j,h,n,b,f,g,v,c,x,s,d,f,g,b,v,c,s,h,l,p,o,l,u,y,h,g,v,f,d,w,a,v,e,e,r,f,f,d,t,j,h,g,a
16 42,erfe3,didsf,rfiame,524,qnerqe,fdfs,f,xxxc,v,a,af,fda,wqqr,adasda,xcvxcv,sdfsdfsf,ewwe,y,t,u4,524507,ry,cn,b,sd,ef,rtq,gw,sf,sg,sfg,sdgd,fe,sf,df,a,asd,c,a,al,defter,kalem,kitap
17
```

Analysis

Stack veri yapısı LIFO prensibine göre çalışır. Son eklenen eleman ilk çıkar. Bu yüzden dosyaya yazdığımız veriler ters bir şekilde görülmektedir.

Burada her bir Stack 4 satırda gösterilmektedir. Çok fazla veri ile test etmemin nedeni sonuçlarının daha tutarlı olmasını istediğim içindir.

Yapılan testlere göre oluşan zaman farklılıkları şöyledir:

StackD > StackA > StackC > StackB(çalışma süresi bakımından)

En hızlı çalışan implementasyonun StackB en yavaş çalışanın ise StackD olduğu gözlemlenebilir.

İlk kullandığım StackA ArrayListten extend oluyor, ikincisi (StackB) ise ArrayList i direkt olarak kullanıyor.

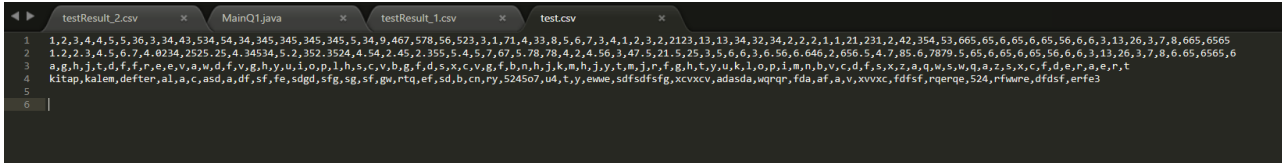
Üçüncüsü(Stack C) Node kullanıyor. Sonuncuda(Stack D) ise Queue implementasyonu için LinkedList kullanıldığı içinen yavaş çalışan odur.

Ama çok az bir veri ile çalıştığım için süreler nanosaniye bakımından olup birbirine çok yakın sürede tamamlanmaktadır.

Q2

1. Test Cases

test.csv dosyası:

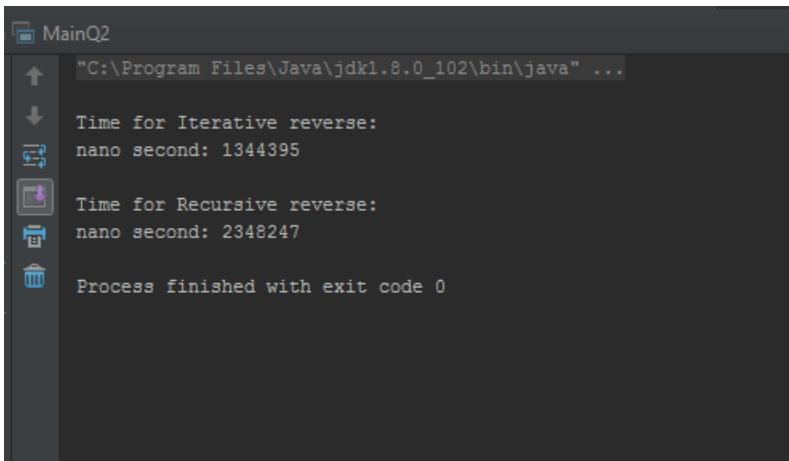


```
1 1,2,3,4,5,5,36,3,34,43,534,54,34,345,345,345,345,5,34,9,467,578,56,523,3,1,71,4,33,8,5,6,7,3,4,1,2,3,2,2123,13,13,34,32,34,2,2,2,1,1,21,231,2,42,354,53,665,65,6,65,6,65,56,6,6,3,13,26,3,7,8,665,6565
2 1,2,2,3,4,5,6,7,4,0234,2525,25,4,34534,5,2,352,3524,4,54,2,45,2,355,5,4,5,7,67,5,78,78,4,2,4,56,3,47,5,21,5,25,3,5,6,6,3,6,56,6,646,2,656,5,4,7,85,6,7879,5,65,6,65,6,65,56,6,6,3,13,26,3,7,8,6,65,6565,6
3 a,b,h,j,t,d,f,f,r,e,g,v,a,w,d,f,v,g,h,y,u,i,o,p,l,h,s,c,v,b,g,f,d,s,x,c,v,g,f,b,m,h,j,k,m,h,j,y,t,w,j,r,f,g,h,t,y,u,k,l,o,p,i,m,n,b,v,c,d,f,g,h,i,z,a,q,w,s,w,q,a,b,z,s,x,c,f,d,e,r,a,y,e,r,t
4 kitap,kalem,defter,ai,a,c,asd,a,df,sf,fe,sdgd,sfg,sg,sf,gw,rtq,ef,sd,b,cn,ry,5245o7,u4,t,y,ewew,sdfsdffsg,xcvxcv,adasda,wqrqr,fda,af,a,v,xvxxc,fdfsf,rqerqe,524,rflawne,dffdsf,enfg3
5
6 |
```

2. Running and Results

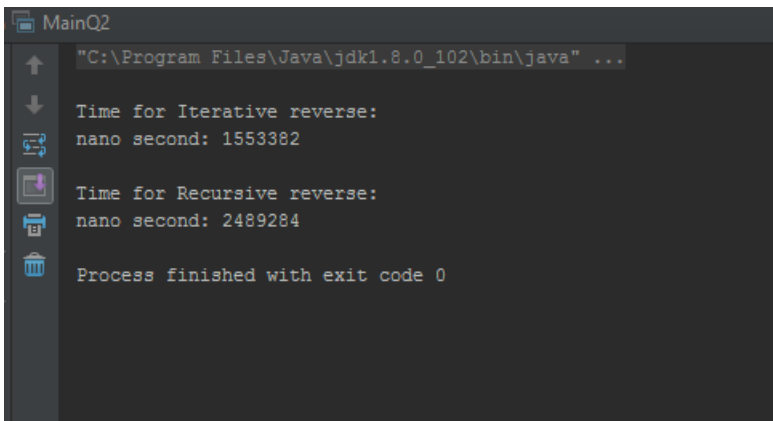
Ödev pdf’inde belirtildiği şekilde test edince zamanlar şöyle çıkıyor:

Run 1:



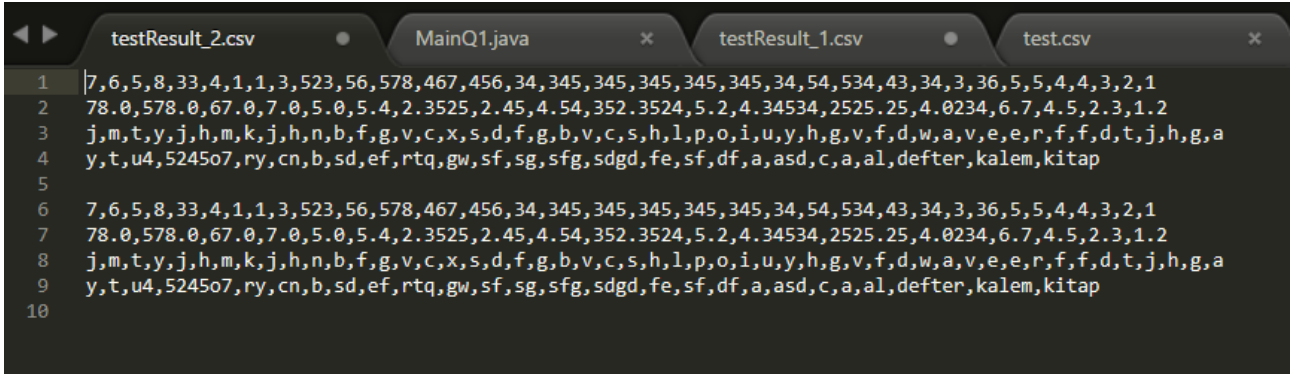
```
MainQ2
"C:\Program Files\Java\jdk1.8.0_102\bin\java" ...
Time for Iterative reverse:
nano second: 1344395
Time for Recursive reverse:
nano second: 2348247
Process finished with exit code 0
```

Run 2:



```
MainQ2
"C:\Program Files\Java\jdk1.8.0_102\bin\java" ...
Time for Iterative reverse:
nano second: 1553382
Time for Recursive reverse:
nano second: 2489284
Process finished with exit code 0
```

testResult_2.csv dosyası ise şu şekilde:



```
1 |7,6,5,8,33,4,1,1,3,523,56,578,467,456,34,345,345,345,345,34,54,534,43,34,3,36,5,5,4,4,3,2,1
2 78.0,578.0,67.0,7.0,5.0,5.4,2.3525,2.45,4.54,352.3524,5.2,4.34534,2525.25,4.0234,6.7,4.5,2.3,1.2
3 j,m,t,y,j,h,m,k,j,h,n,b,f,g,v,c,x,s,d,f,g,b,v,c,s,h,l,p,o,i,u,y,h,g,v,f,d,w,a,v,e,e,r,f,d,t,j,h,g,a
4 y,t,u4,5245o7,ry,cn,b,sd,ef,rtq,gw,sf,sg,sfg,sdgd,fe,sf,df,a,asd,c,a,al,defter,kalem,kitap
5
6 7,6,5,8,33,4,1,1,3,523,56,578,467,456,34,345,345,345,345,34,54,534,43,34,3,36,5,5,4,4,3,2,1
7 78.0,578.0,67.0,7.0,5.0,5.4,2.3525,2.45,4.54,352.3524,5.2,4.34534,2525.25,4.0234,6.7,4.5,2.3,1.2
8 j,m,t,y,j,h,m,k,j,h,n,b,f,g,v,c,x,s,d,f,g,b,v,c,s,h,l,p,o,i,u,y,h,g,v,f,d,w,a,v,e,e,r,f,d,t,j,h,g,a
9 y,t,u4,5245o7,ry,cn,b,sd,ef,rtq,gw,sf,sg,sfg,sdgd,fe,sf,df,a,asd,c,a,al,defter,kalem,kitap
10
```

Bütün queue'lar bu veriler ile test edilmiştir.

Analysis

Queue FIFO mantığına göre çalışır. Yani queue ya ilk giren kişi ilk çıkar. Bu yüzden dosyaya yazdığımız veriler ters row ve col da basılmıştır.

Oluşan testResult_2.csv de ilk 4 satır Iterative reverse ü gösterirken alttaki 4 satır ise recursive reverse ü gösterir. Recursive kod MainQ2 nin içindedir. Iterative hali ise MyQueue dadır.

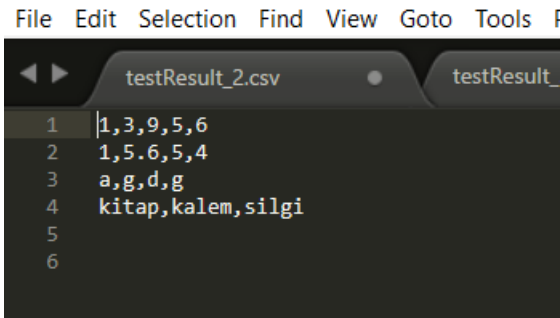
Yapılan testlere göre oluşan zaman farklılıkları şöyledir:

Iterative Reverse < Recursive Reverse (Süre bakımından)

Görüldüğü üzere iterative bir şekilde implement edilen reverse metodu recursive haline göre daha hızlı çalışmaktadır. Bunun nedeni recursive hali quedan sürekli eleman çıkarmakta ve bu çıkardığı elemanları tekrar tekrar eklemektedir. Bütün queue ya bir kere çıkarıp bir kere eklediği için çok fazla işlem yapmaktadır. Iterative hali ise sondan elemanları alır ve elemanın 1 eksiği kadar elemanı baştan çıkarır. Recursive haline göre daha az işlem yaptığı için daha hızlı çalışır.

Q3

1. Test Cases

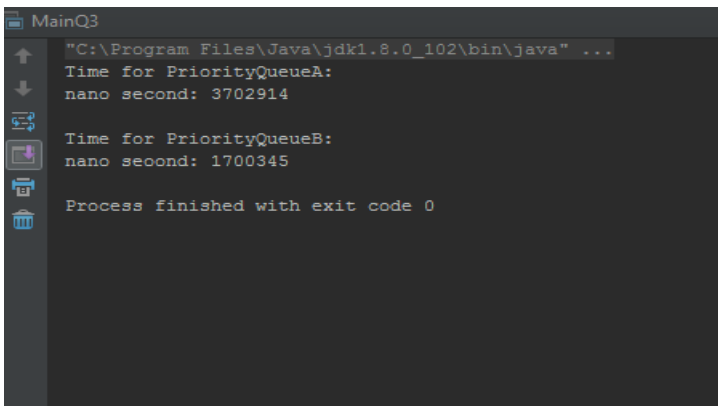


Bütün queue'lar bu veriler ile test edilmiştir.

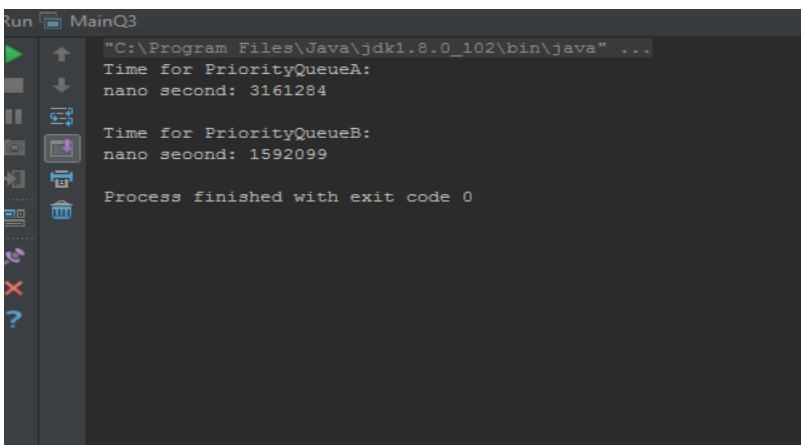
2. Running and Results

Ödev pdf'inde belirtildiği şekilde test edince zamanlar şöyle çıkıyor:

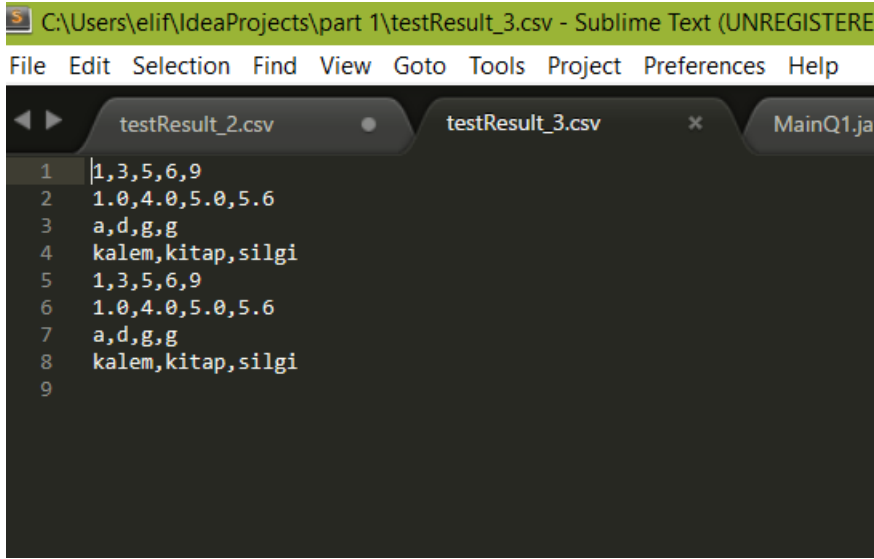
Run 1:



Run 2:



testResult_3.csv dosyası ise şu şekilde oluşturmuştur:



```
C:\Users\elif\IdeaProjects\part 1\testResult_3.csv - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
testResult_2.csv testResult_3.csv MainQ1.java
1 1,3,5,6,9
2 1.0,4.0,5.0,5.6
3 a,d,g,g
4 kalem,kitap,silgi
5 1,3,5,6,9
6 1.0,4.0,5.0,5.6
7 a,d,g,g
8 kalem,kitap,silgi
9
```

Analysis

Priority Queue da her elemanın bir önceliği vardır. Elemanlar giriş sırasına göre değil öncelik sırasına göre işleme alınırlar.

Result Oluşan testResult_3.csv de ilk 4 satır PriorityQueueA yı alttaki 4 satır ise PriorityQueueB yi göstermektedir.

Yapılan testlere göre oluşan zaman farklılıkları şöyledir:

PriorityQueueB > PriorityQueueA (hız bakımından)

Bunun nedeni ise PriorityQueueB direk içinde tuttuğu linkedList i kullanır. PriorityQueueA ise LinkedListten extend olmuştur. Bu yüzden daha yavaş çalışır.