CPR E 419 - Lab 3

Anjana Deva Prasad

Experiment 1:

**“Sort the data in “input-5m” by keys, using TotalOrderPartitioner.
Use up to 10 reducers.”**

Command:

```
cpre419@cpre419-VirtualBox:~/hadoop/sbin$ hadoop jar ../../Downloads/lab3.jar sorting
```

Class: sorting.java
NO Command line Arguments

Sampling parameters:
InputSampler.RandomSampler(double freq,
int numSamples,
int maxSplitsSampled)
freq: 0.1
numSamples: 1000
maxSplitsSampled: 100

The following output files were generated. The size is indicative of how well Hadoop’s TotalPartitioner was able to perform load balancing.

```
cpre419@cpre419-VirtualBox:~/Desktop/lab3_output/exp1/5m/output$ ls -ltr
total 278340
-rw-r--r-- 1 cpre419 cpre419        0 Mar  6 19:52 _SUCCESS
-rw-r--r-- 1 cpre419 cpre419 30201165 Mar  6 19:52 part-r-00000
-rw-r--r-- 1 cpre419 cpre419 27744750 Mar  6 19:52 part-r-00001
-rw-r--r-- 1 cpre419 cpre419 27040515 Mar  6 19:52 part-r-00002
-rw-r--r-- 1 cpre419 cpre419 25479684 Mar  6 19:52 part-r-00003
-rw-r--r-- 1 cpre419 cpre419 31994100 Mar  6 19:52 part-r-00004
-rw-r--r-- 1 cpre419 cpre419 28690608 Mar  6 19:52 part-r-00005
-rw-r--r-- 1 cpre419 cpre419 29331402 Mar  6 19:52 part-r-00006
-rw-r--r-- 1 cpre419 cpre419 28824216 Mar  6 19:52 part-r-00007
-rw-r--r-- 1 cpre419 cpre419 24942972 Mar  6 19:52 part-r-00008
-rw-r--r-- 1 cpre419 cpre419 30750588 Mar  6 19:52 part-r-00009
```

The first and last 5 lines for each part file.

I generated this by doing a cat followed by head -n5/tail -n5 for each file.

We can see that each file has been sorted based on their ASCII files.
Moreover, we can also see that none of the constraints have been violated. The biggest keyin R_i is still smaller than the smallest key in R_{i+1} where i can take values from 0 to 8.

0
0001MeE7B7JPCWj Hslz3NRPzxugmQlH9s1UW1ku3jPc79T73HdtFxGJ
0001lcFyJCuKl4l vRlpXC3AuprUAMabMLgsIeSZxURRHKS0mi9VprSX
00023bbrDdDCyzn Un47nRNdoCLyDoh9irbhbRDuxIRgxB4HctA7Wubd
0006rvR9W2eYJL5 87fUZNdKD6C12lFoDagEcrJxZXce4sPTn4X57Qas
0008P8NCnnLVhJh qmRAduMXpSRodW8PCiPDtCqc5oJqLCuJaix9YFLp

6SMW0BH0Id80FGK Ik3qoeSlaEBczrhIALOWJ8FcoGEM9bTpi985BEys
6SMeXV3Dvo8LrIY 02volVCBEcACHa01SsGtNryvuY5NFKKU5UQ2b0bI
6SMf9LKeRXJts66 XmontLhpUiontK6oxeKR83L3X2hdtFbg3VextOSM
6SMmRrS8rece4AJ r3NrE4oSAveBVVT5NtrZmcqgiCiPKi8URMcaKq2v
6SMn8WMqLVyemyI YLpV3N4qbHQbs5ByDmkJvu9TpoCb6n5funkch6Js

1
6SMnVeGiyb2vmIJ hdkMub5RUkzkENMZkR15AFm1uEBSAl3rq8BpoZzs
6SMp39HEUKvBiei eTR5SPdqHZ9jvcptpbQVGk4iYMpqF53PpECrTNmS
6SMpZK6dBBl88uh 9oqkOcs36bKf3VPAEBxFiH1dNiEolaLcOOc7O3Bd
6SMsHm7WUOPWOZp Y90FCff7O0SHXZhL2NtHCUr08IYbMM5860UR33uf
6SMtcP7cu6o675s 5FPjOsOzRUfd5qe3PkF5YWUSGhdemCgMRv8x1zRu

COz3tLH4KgpVArH e6JD95p1YW0eFOpeBCI3BY9O2bl5As5iJz8dSFxg
COz3zea2AfrPKMZ ziijD2TD22YBnWNCJf6cNdOlC3emtFxrW28Q25UY
COz5512iKQWxO38 IFTjmuVMifiqRKKqbFsNazPQIcAMSDNrN6po1UWf
COz6FY33MgPKr81 WtMhKLato40kno3zFniBbX00dYp9SAFU4WcWIrVO
COz6et8kUcm56i8 SLAxFPH396lRuik8RNsTJoUrm1mlAso11N4TQ1Ev

2
COzCqeLDSxAVOcS BO0BWDrntWjpxgbyuJHTV7gHalNJXyPbNkmIsq4h
COzD1PlZj3Ru2xv 1j0SNp0qnXSt7ZakAnpU6XGP1DxuXAvB0drB45Qj
COzG9A27l9XnbVa a2mbpeVQziJ6lbIXPgGLIzIHJKKydVXx3L15s11F
COzHJ2oFZKeN7rj tozkalYiorLy6ayvTAssbHQJBZDoOm5Y7RXOsBE8
COzJjVN4ydv9eCj GeGba65UbnhcKompB9xJhA0iNXCnTAjoIl9APAuG

lBVPAusR12JW3UO aj28ZuI5an11VmAXDUqy9AOhvy6IEzNu8W3qoqfh
lBVVFNbgGVf96SJ 53YElrEbCK1RQKC2Xnq98CIapyyKCfUm01gWXOMk

lBVXejuJcCV2TTk kvsp9ryxJKSuku5MBqyxJ7NtM1kz1NUMBBKkSmCu
lBVYkYd6FxSWBMa byl6ARuYzB2llf0cjbnynaA2PgcYlQ6T5pWS3nEl
lBVbKKxqLPt9KCb lqHqigAmaYM9LF0mHyCT5Sj6X8lNAlVYJPB60SCN

3
lBVdOH4UeLqLUxh 4Fh4RWbkiaC2HUHBjsrYptIGdKFkvefSuuUSPT6L
lBVeg1TmhhJxr24 ALd6uhqo1PdpE8YLCXuhM1JZ1ZTDFXeS5uVaIr60
lBVqtDTHxzKE6FR oKjG2MiTYmQMpnxb99CvMnlhBoq2UZoPHQfmbsPT
lBVx1tZESzbNlAV cNypmUVKTOWNdgYGF0lhXRxlg5KK4pDB2s6aliJ3
lBW0UZiqXSI9XuL vRjjO4OJ1zuBDGQUWj9drWagvSGCOpLZhd5zUMx0

NdEVSe0FUYq9Vf9 J32WVPQHFxnRnYtGmtDYM1S6o1SugzecbJMe3pjA
NdEVUCcF7JAQGH2 5OIxDYGF2zpLv3GK8VyPuMz1N3dvHrX0noMONLEE
NdEbd6crCm2OuCb tdRNOlCzxFNmqdAv0H448nHiNUUNLcFbtzlmEUD3
NdEfBAgzONRTV5W 9Yln4BNztnAV149stlAOR61O7jAd72MOUyPiDc2i
NdEjDn4srGL6l0i L2Ji6Q48jFL4Cb1to7OXPX61H49rrVE9gkMrkDto

4
NdEosVBxjfWcnRq X6kKb6P6q25qaZrekpX9t2dCWV2zLRjkuNxDGOdC
NdEp8vFVFxjkxMb jfai0vXGUsxirbXg3G99Sijrn3zBh0rN98pzGz3v
NdEsunaETyk12ZR 25HT0IepsccXxQuooA68z18ZAEjk4z79uXWNoUCe
NdEyx7MioCuUqdP behe6QtE0GCF4euE5WGZJ5h8s5QFSadWPWu6Z64H
NdEyzA0OvHBIlj4 4dt60lyjjbqsSkKnVp0Qzi1dMFK6tx2GYTXmzhnV

UULop9d9FU73YQS mKkfV9oWJABHN3AvPKfQpUgQs2S8WPYjhmEHHDzF
UULoqfLmO7hkDAk uOCX6QTrlu4k6Az9nWNh4kkUMjjfDqgrkL29yCKD
UULpgD8fyhq46kd tubCQtVjdoTz9A8BPS3SxxnjWPWbX6G23aW1jyHK
UULtetMKOzQUtFN N5D1i3cGsob8HcNfllVSf75IcCeTO637nki4gPBh
UULtqSEG6lW8B5Z ll7HCOO5DzIt5ipyT1TJO7QAcZ3Rk2Aj0dFHmUum

5
UULuNX9d1O8b0qc aL0LAXdXbb9N3pT9BvVY048TaW1uVkPYYsUTn2SC
UULvZfffsQSuJDy GhqZ4mCNlP3yAN9YfLtjTj0EljxAr9cPigLb6sYJ
UULyP4XzlL7zxlO 21iYKUZldoXOrRQox8lVpkQkdrqeVg2DpbQyj5cr
UULyeuRccfnqMgC ENUsFsKtMS9DiNbaSg3N2MTai6rlNdJ7NSMz1bIu
UUM2tNGXmXKaUp9 rOr10lnkbFRpAZaRZJfKKl28lIliaZ4Ex5ujmpX5

ac6fafHsuetb6lA Z0nOy27Eza0mI6Rovb6gpLsFTPj7QER22jpdzne6
ac6frbyddjpPRIC 1xnOYu5VLZJr8k83AK4BQTJOinJvt4YjiYSgBFFO
ac6h994fmNPQDOG qCt7b6McfelKhWvoVpIgzGIn3udWy0n07yWxt90m
ac6kRI8neCm2MSu AXKRf0HvkE9fOukeqUJRIJJC5uOM1LO9pmWjLQ7y

ac6qaQNV5II8WCT ZVXFMzq2yT1Dg9nCt6t6WdFIp244bMt8Cj77ifyd

6
ac6xFQ2bLlaaOYq Lrk7yAZeNh2Rgvtuv0C3oaddvRSGgXTYBqxSFQfK
ac6xnjeLqcnb5F9 X8UbmYcxzxH85Z7TYm1HGNLfMpSL9zGtXOQ7prFJ
ac70g0URJ0L31Uj Sa1ttqnFy9SrRQFAK8IKWS01UFcxWeuoZAN4TI0M
ac73V2lk6334jbF XhjxjOfUUHIMkLJpa55hCUDAAvQCuVQIn4hHczzg
ac74dgPdjKLyy7X WtHuIPqj40Q7Zj8HVVqZoTZkzrlW5ksnc85jbUL5

gsgncMKUuL8jM85 lryj3Zz3atuYs2BcvDkq6jnJYluXEyIrEeBkgbrl
gsgqGnEZHQt7v94 NIDUOZZeZINt5TAfI0B6scTmOrgOWpNqe2rOfEM7
gsgt3Hvc2hEAg3q 21Yu3loabqfVSf5t2dDfuqYUoFBaC8v4SXygCWuG
gsgt5BHFALvrMc3 GRGWc6JIZbiDfPNJMQLMW5nHAxqNLe8G9euCzoNd
gsgv36uxSARG6Wp xbh9ZpCKuVcaoXfMIB7z8HQEs0vfWkyOSX6A2JOI

7
gsgyKPqabMuNrGJ Q5CSzMqbUKedAzpTGBVIHbffgd9AiKSGKfFsUCFB
gsh4VFm6GApm2my bj8BlW6zCiZyT4kI4VE4CUuYoCWMAAeMjhmH1Guk
gsh5WxnlBfGuD8t k0xPuGoJjWE0TFhL4WGv8MV7Tt6UQMyAAKMNLbWq
gsh5hDol8cXXl7s kbjqkVLmVTJYMeFFtKoSyzGHTslb1WHgEjtxKch5
gsh6XWZsnNiCz3D ycd3md7bQJWU6iLpqzoENCxP6jXEXzk7XU2qFFC6

n5K3iBNHFINmYOv e2ZzR7CTyff6QKTvcNRJJWCRyGs32iXy6ohRjNiC
n5K4sBnfVngIvJn FiHPeYvlh2PuEyzJGzWyGZUtV42Z20ckUWlfpTJH
n5K4ub7Xs3SHTPa MA8sKEhtxIx99XYEBjr2FOeNWqI0fxkGLLM5PquI
n5K71e7g0BK5K17 aquoVdmmMHZDzZOtluZUAIlvRUev7rTGphKKkPZd
n5K9DRYO7Im0lOl kfiBybi4dfEEoXlX1WKGzG7x9AdTpCmKS5XH4rhx

8
n5KBvtTZKogooqD BsuBS8eSCeo21oYtqaxNILld987BA5y9hYg2nig5
n5KEzdGmW6WgUam x3RRaV8zx9YCVrtk4lfjNH3NGxzI7Jf1t56lKWBV
n5KGNiBvWuC3ml8 aZtYRKkCtP2mcU8qQkt0ePrpqBAmeOfsH7YAtK03
n5KHFXXaf0Fc6xp WftxbTIJY9OOqarKtpuTWLdrxNtqnjiK0AXIk12U
n5KIkJIGU9MuWJY seXW1kQSjSdtjxTrXtUYy8u4z3hUsVyvPj7rNyOT

sQHebmjZ9DsM8tM mBrs5Q3k0DMLgN6EK9jVhqlkFhOdMghWrD66teAH
sQHlZltIAadPUFp WIaJm0WZXkHTu5xgSxlnVtWAipIaGDJH5LdCA5AN
sQHs9tacSJ5lgYD 8BQPTGcEVeimohvGyxfsnvS5faiSUSXRh9Z4aVbq
sQHxnFqKAiZ7vg8 8avHFhx7DofVSFICcmpAWrNe6vJokf0L8t2p8bYu
sQHxzmxB4YI8nvQ 5amjvUeRoQrrmfiijJiUCv9cWCccHRuqlSQtdqWq

9
sQI07EIPvz7KT1t rt9JeuSLfBFn7YITH2CghWndEa0KBxUsy3qPzS81
sQI59icVtJFn867 f8LnTI51ArVuJHzGovtompv9fyeLi1pPhgy7InGM
sQIJlkD9hiW9Z3N Rth8unZBWngqcn3XWSpyJgU4rleExxtNRAQX2h4e
sQITMUCxSgjmFBT bQ63abNutv1THvs99nkSWpLBTZyIVYKVMBRF2V34
sQIY717C4Gg1PqW UT6DZmN2Ro96h5ZID6hoZx3ehPHsqink0q3EfaJl

zzzlsNEhLA6XXr4 358YpGifuI8Wvb0f917CLkUTmcIMXn9UFpWFAiNm
zzznxiU8di17hBk zMz4cONqHlhp8PMnHaLtU7VB2pDlrS5Oo5kKYbip
zzzp2osLN7Didmk NA759V0qkWGK2Nfd8EvPc5FL9FgAIUbIX52JR4o3
zzzuoUL6N4qMAFu JzmJyyjISlKaSWPi5ObPcZoT9hEtHg3VDYQPTIco
zzzz0yxLW36DFE3 U32iJXn1rJyHPFOCvXEObCDsLT2pgv1IdMqBaeFI

Experiment 2:
“Write your own “MyPartitioner” class and use it to sort the data in input-5 00k. You MUST NOT use InputSampler class. Explain your algorithm/strategy to partition in the submission. Use up to 10 reducers.”

Command:

```
cpre419@cpre419-VirtualBox:~/hadoop/sbin$ hadoop jar ../../Downloads/lab3.jar customSort /lab3/input-500k /lab3/exp2/500k/output/
```

Class: customSort.java
Command line arguments: src, dest
Src path(Dataset location): /lab3/input-500k
Dest path: /lab3/exp2/500k/output

Sampling Strategy:
I choose samples based on ideas borrowed from sampling based on tossing a coin. This is done in the Reducer phase of the first Map-Reduce phase. Using this strategy, I collect 10% samples from the dataset which is further sorted to computer the boundary for partitioning the data.

Coin Toss - Random Sampling
We randomly toss a coin for each sample (programmatically, this is equivalent to calling Math.random() and storing the result in a *toss* variable). Now, based on the outcome of the toss(the value of the variable toss in our case, this can be a double value between 0 and 1) we can decide whether the sample can be chosen or not depending if its head or tails(programmatically, this is comparing the value of the variable toss with a threshold(say 0.5) and discarding samples based on some criteria(for ex: value <

threshold)). Thus, as long as we've not collected enough samples, we call Math.random(), compare the outcome with the defined threshold and either choose or discard based on whether the defined criteria is satisfied or not. Once enough samples have been collected, we can stop looking at the remaining samples. The number of samples chosen, threshold value and criteria decide whether the performed sampling is representative of the data or not. In this case, picking 50000 samples for a dataset of size 500000 and discarding toss values that are greater than 0.5 gives good results.

Load Balanced output:
The size of each file is indicative of good load-balancing. The input file is 29MB and each sorted part is approx 2.9M(29/10)

```
cpre419@cpre419-VirtualBox:~/Desktop/lab3_output/exp2/500k/output$ ls -ltr
total 27856
-rw-r--r-- 1 cpre419 cpre419       0 Mar  6 19:52 _SUCCESS
-rw-r--r-- 1 cpre419 cpre419 2811240 Mar  6 19:52 part-r-00000
-rw-r--r-- 1 cpre419 cpre419 2811468 Mar  6 19:52 part-r-00001
-rw-r--r-- 1 cpre419 cpre419 2871261 Mar  6 19:52 part-r-00002
-rw-r--r-- 1 cpre419 cpre419 2823438 Mar  6 19:52 part-r-00003
-rw-r--r-- 1 cpre419 cpre419 2846979 Mar  6 19:52 part-r-00004
-rw-r--r-- 1 cpre419 cpre419 2883972 Mar  6 19:52 part-r-00005
-rw-r--r-- 1 cpre419 cpre419 2907285 Mar  6 19:52 part-r-00006
-rw-r--r-- 1 cpre419 cpre419 2826858 Mar  6 19:52 part-r-00007
-rw-r--r-- 1 cpre419 cpre419 2855529 Mar  6 19:52 part-r-00008
-rw-r--r-- 1 cpre419 cpre419 2861970 Mar  6 19:52 part-r-00009
cpre419@cpre419-VirtualBox:~/Desktop/lab3_output/exp2/500k/output$
```

High-level end to end flow:
In the first Map-Reduce phase, we compute the boundary values after sampling. The results are written to a temporary file. This temp file has 10 lines where each line gives the upper limit boundary. Once the first MR is completed, we read the temp file in the main function, and store the results in configuration(conf.set(key,val)) which is used by the Mapper Phase in the second MR for partitioning the data.

The first and last 5 lines for each part file.
I generated this by doing a cat followed by head -n5/tail -n5 for each file.

We can see that each file has been sorted based on their ASCII files.
Moreover, we can also see that none of the constraints have been violated. The biggest keyin R_i is still smaller than the smallest key in R_{i+1} where i can take values from 0 to 8.

0
000ckhLmtS7AeAE 4D2NVCzvxJxrXCeX0NIXDrVx2gxNQp3knjt4rVzv
000dJJ1TKvD9JY3 qyEIVmbqdmMQi27871R6OncsDoJRAMxp0MjfEGtW
001SZ6pXMRQSVxz vm3eElvShsy3lkQ6MaiP7UejGmq1Rsu2dQl2VdmM
002CGxZhgFpf49N 5MVBj8BvxZ0FehM35RlG6ZdL6D0TVeps4smgONDS
002DLQWKhfVtQPz ItqHFG2F0Het6RZUFODMRj2s9MYhXJFgj3Qu9Oqc

60DhfQr3ZvU3zoF nuaxeubxssmOFnxVuOZhXOlUO9G5JiAlMOpPiXHG
60DvUS1J20ZngNF 38ApED5TtSg1CKlz48774zaEL9qTkvYW7iRDlgNJ
60E8MvMFGPcmqhe NV0pjPOJSs6IIsA6YlraMXyZb1rFH3n5AKTTCIa3
60EBeXcVX3JWm87 ojc0S6jxbQB2KZ1Q6fGoKIT0Idb5RCUbzXi7eg2Q
60EHSVQL6h2aljf ZboWosup1smsJi2kNOYjf3tm2NClGja4oHa6qRfu

1
60ETr3LZtpsbWTz 1fryGjRcKFFbc6aSlZy4c8RUFtcExAJaEEp9MFzF
60F050xtCQQVGLz XtyODCnXGkJIW99IdvMINSF7AXCrWG13TLdM8ckE
60G4ZaiZp10UBdi iGiXR5MQmS1oTnS4W7G8gkb77GXGASS6Uol9bHau
60HFnL3aImjf3ZU 8mzttY5j3IqUKVc5B1pJNq9LmCREEBeCNyvg3Iv3
60HKznb9TSo2yCE rusMb061Rb4S7HHy7PziWiQHmNUoFUB1pm3pB5aV

C0eJ2H9zOH70ECZ lmbTSWrQKlNLpGl1OVer5xfn5sCDF1dGEReKRocP
C0fByQit8WjeTM0 pQhBLlLgNTyHsCn2GlyaPoM5A19CDNlrfMlT2A7h
C0fnBPJv3azCO5M BRWt98i60sbAvXObdWOrDzvS5tPdxjR5idccsjz3
C0fnQUYMQ1CGrGd bAiHep6mJ6sVvrQr6G2s1jv6kDlmaWLlUxyyh384
C0g0RddcP6fa7CA A0y7XTHx8vo5PlToanHqnn3VMFM1zO0MUHgkZrtr

2
C0gQMXpMQjXaOgZ 3gE4mBrqSWUtNo4VU1CuKEfhJDQX8FsT1YoKoHd3
C0hQT4Vfjksz1Ye UeNBen1oqiBWPFdf2VYGsDRPcnSAf4qhySV4UUVN
C0hVZWlfHYVM0k2 T6q4WygXpzS3EPoCsNZlczexLl7iOoY1sAPVWtjR
C0hgJQ9TFR89G9t W4kimFQA0DbADL0ifx5o0Yq6y5OLmUJTnx4ctHSC
C0iomRXAskIxCCe MnZvIfBZebNMkvaJCv1ATVyl0DG61FzZBNjYk2rY

ICdA6FTN4dT6WT1 H6ex5aMzqUTBa5SHOi1xxqr0Jl5BH0Ap7eC4sU3n
ICePythBPrAaDOJ sPcWnUtQky8iYjg9Z1Al1oEK8afHOlihmU1FVbFB
ICeevGDv21fOiQh x03QbcNUqRsNuCC8ajPF2oDfaHACgB9HLVfN9iKH
ICgGSOZUgci8vCd 14sLAfZ6xjhrCXE9V4tMCT9qEaC4ja8nuTJyc0rm
ICgJt4n0NiiQgbQ sli6zubZ0hM9Q84NMQ1WMTQzLPrsckKN73q5lqTJ

3
lCgLTJjn6QJdxuv 0bYhVRSQSCXQ99V3NtKS2vgDiCLgayYH5NYWqiOL
lCgmJLNxfhYJE3Z VIyz6MY4v6xdBHWiGK9fHtaFChdh92l7u0EF99TV
lCiQ2vxaQOsT3BR 10oKaXjHPhxPfUED1WgNyLo7qKOUSXO692uO31Qq
lCj4LKRpJVCnOfT 6BXLNgGMkkb5lY3lVH5YqyY4i9ST41fMmgFstSnU
lCjnciKGgV1QqJZ bhj3JMOIAFVa7X7cqPBtpvocn26fkLVaL8jOm0WZ

OFFR5VvxE4JikVq eEX0dOr7HqfMp5pYaqvEMSzZLkvEzRbvC0vWo1Jc
OFFeYcJTlazFixj eOqyHCJXgdF2fH1nSAvaj6AcsX939u7N2d2W8OGM
OFG7vdEObZEFWOf aAZEHt6GxHJJ9skCO9gU3eWShsMu0aessJz16nQW
OFGx8qXGoJvqDxS H5ymzdmybpJoS1r44JkvDxtfK7U5pRACvKZ0QkFa
OFHLsd019tPh4ZM x1AZ41B8Wsot6UaH3KURrFqHt0vxA7iYRE7kSKWm

4
OFHbrrZgdyKHBAn Xs6TNL1SdGJAk15gFlyCq91S8fdBuVOLJYEt5mRk
OFIzncsDWfiX8SE 2X8fWZOdkq0v8cePlkTheBnnfp7NI69MnI9BNvrA
OFJ0EUU8hD4SM4k SmmkPSGQLK1s5HI44NPtycYIladc3Xyxxt9EcY2K
OFJHstFHhxYIUuX R9nsQKkriMn3GN7oAAc3pxODT5yGcNPacob26Y06
OFK6E2IkYczslDs BiPuqOPAEXiTSfNyexxFzQG3vAZWmfd0PeGWCHLE

ULUaOfl0p62PQlP ALWyu1ui0HxHN5rX0Y40LQrGNYOzPhZHAMuzdmIl
ULUqJlEgbQqlYRh 0Mi3Le69VbedpjtPL655571D5A8zpDENlGkG7GQ1
ULUuMOLYQTIcmsL OkHnbcb5Kz8ovcCy5QvykiaslppryozuzyfvemxW
ULVLiGg1yKX1kql rqMo4ODOY0Fj4sLq4zEbEntqTYZtAIa3o7BVdfQS
ULVbFnRse3JKt9E zbnKRdfYkR2RVc1gdCbSsn5tOH3WLfyjkmMmtyzD

5
ULVetiGyYLIcGRV 1fTeYCp3xBkJJmcVooLp24lQzdEdkRPR7uIhvWeD
ULVgbZvG00Qqk6F hVEF5truOXZDMvfP13sRtWFFVFKN0PHBNXeZIydV
ULVhDa9JUxLqVgg qeq7axUf57sAZsUeZYmGGXzgnNvm9keISLOKd4fF
ULWGCvSc5NRV8Fa 05nY3x1r5LYQkFoBo3OAR6qjlv4A3PMM0fq0GgaI
ULXHD5177DrnqAb ax3ngLceA6Ft51Wo42R6nrX5RnmmcQSpteoAUiol

aXVWZZDkllQmj5q x54fJ9cahsJo98ka0fE5SZ2LrG4fjcPNKp0fBuKF
aXVmTg2x8FMFrVG tEGKpN2t3z1HnocGjSDovVUL6ZaGIQQpvEWDR3Fo
aXW0DLnFuVxY7l8 stnru3FVYyBqUPjpE8c4zlnJ2v6fKqBYuQrAGX7s

aXWMI6JunCA0TXP UHCQRRaNLp8aE7Or71it3DROi7iXfAbzXZz5kuot
aXXT1gJnVhTAjEj UJbIaFr6FAQNEusbcGVifTREzt4RZ60Om2vLMSOJ

6
aXXeJAVbvKjKp7V 3zTZVZfsJGYeM9y9O4u6K7GtNlGNZdZl5Sij6nXv
aXYXaIeOVdhxUes DSHrTPJSagn0xPmudAuN03YZDJBZcNOHcX7zSdxQ
aXZiaMOgyJa8bUk qlW8Z8K8RWkdMsgjJvG0Tq3afZZsuKUvCNnLhsps
aXZiuAfqZFuf6N1 GGSD2QPM7NYPB3QWRbZQfLYpchYLaTxSiJTikFCf
aXZljylxH02qxpX xIsGnFfsmAK8h11vMpZm29irQfR9sCQM6TC9QUFP

giYyMdOpS96drpS nmJvLyKShhNdx18qIU9DqqUFWjgUeoAr5cCpY4Tj
giZFtNZ1urs61zg 4xoikRul5f2DYNxhDUnrR0EId1q3NB3pbKYVUzMV
giZixRZbRfs2ex4 EnWLBc1DfmTRexgPgFSqW3rYKgpvqEX1YU5OjQNu
giaqjn129BWYou0 dmieuSMajt08zjXSlEJX9FNWl1GJBY4FDH5vS5ht
giarVuCNtiEMO8y XOAbeHSZX0La5GaCx0AmHWFnae7ju5cU4hqK4FB2

7
gibbqPBXABJLclX sDZyo8Pq0kyhiK8D1042gh5NMTJYefVFvsPgCH96
gibjHzT85yPJ8q1 QD6FyWB7zmS9EIV4Wo1qboClZuBmoiqYKy3HQbau
gidSZpvJpGMA2TI Sgu2oaXfE4yD3JnAbVNubnDNs9IljxBbQqy3bQUE
gidVyK7JJ710g88 Y70suuq6yDGrjDyc8fR3yGWVcjmBt3a0frJTl8y7
gie7fXm7FG5GKs0 sjdj0aeePC17mVbrafWcC4ed95a2ji2ZrNbj8cS2

mmDKfUFnKEOjLrI MrVdSZhgICCWrYQeVMko8YfFDCl2js4hxmKMnRfj
mmDotpkLKyCcoRo FTIMSmGsMhASYR7FJMWNaBDGKqyzzmWESiT6oWPk
mmDx9T5A94dxF5O jnPVfQ9ltJVnxsOv7xjAULReVVNf44E275gVragM
mmECyaCyE4Tsl7B 224Vlo21GzB89MdWyhVHsKEMze4ASgAnKIv0W46V
mmEGqLOPW9oDCfq GPhj59XVvY2GiYQ6c30DEMeJGjQXAj70mx2Biesd

8
mmEfndnfJMCQyNP dCGeAgPVy5Ppenq0aZx3UfGk4YeRcnElklbbin5b
mmEmFtKnmNat3h9 MDEBPoDjGz3voLdZYQ5V4qV4dH3lvMXQ7HpNr45r
mmEn1COZ2913vOV 3TZ7q4Jad6JSsLcNECqUq2PxXMc1aSLu52W7gsZW
mmEvYacVsR4HEJ7 RnZ4nbYigb1y96zIkvlHsI3q9PbDtDQz125Vmh2O
mmG2CKpmoPNm014 U1S5oH5XttzpETrnT9I3FUGLDl7yoQAVVIO90D48

stOXO42519OK3Jz Mm2v2atrEvtHBlEEfMdqyFO9oDPrHBONF9gOJq7A

stPNrJpLe3hJ5l2 vPNYeOIUJ0h7iiUJuvCulEPqpGpQMj6zXTYp8KAK
stPP5aEEvb5uoWJ ZOJ1EsoZVVJSz4HPgcR0sFLuOIqOioS2MkNzVxxN
stQECjK9WQV1FcL NFyCC8LEf1YuCfnzIpPMdpYjPhCyCoFGH8RcRrmV
stQPNHdWZk0QLGV Xg2LgyOJxxdARfVGEuSxGfSISuZxbPSXrEKb9ma4

9
stQmlsgneskgg25 W5NOyl9ErMYOYs95LT1b73lJbv44QImocYUDo2Ss
stR20E6vgRnHgHa reafevjUtgFm0TGxs5PQgZfclzrIm9MMJ0FyHJkJ
stR786q5xeVQFcx cgSSOFVxC4pc2Ju58KzIeGxP1lHBdIFJuhXt63ok
stR9ifOBkTRjJKy 5r91M3nsdd2G7V4iRIqE6gbWpsIIKYXXYtbDAAti
stRCblqk3yzoWnB jaeaeAnmY84FHHS6rk04N2lxQlhEBKK3c6gbyMVj

zzxZrATXhf7FU0B suICYFi1sUFqLbENYAx8aGzZd4A3xtcPkGm3nJvW
zzxe4LDV1mQeUWp 2q5PF7MzFMCVkSz1EGkWYngpWtOBLkUH4TeXZzDC
zzxuUEUGmDrfLNl DZEWS7p6Rl0Fq72rVbP1HIxr6h7mhEbNJzi2nU8Q
zzyzcFdkM8StXh1 Vg2N3sG7kNcvNMu693xGZTV5BMNPi5vLUTCyyN8P
zzzLyRRmSPdcdgF ql7TSuKJy216BRbO7R7n5M4Mn3FC6TIcijqtycTp

Experiment 3:

“Question: We have a list of datasets (see below), sorted by their size. Try TotalOrderPartitioner and your own partitioner (MyPartitioner) on each dataset. Do you get different results? What makes it different?Which is the largest dataset your solution can sort? You can get the bonus if you can sort 5m dataset by using only up to 10 reducers and your own partitioner.”

I tried my own partitioner on input-5k, input-500k and input-5m. It seems to work really well for all cases. In fact, I’m able to sort input-5m pretty well both in terms of speed and load balancing efficiency.

As far as the results are concerned, the overall data is still sorted in the same way if all the parts are combined and the entire data is represented as one huge file.

However, the data that is stored in each part is different. This happens since we are using a different sampling strategy which affects the boundary values that are chosen. The parameters for Hadoop’s TotalPartitioner and the criteria which we use to sample the data decides the bucket size and boundary for comparison which isn’t the same.

```
cpre419@cpre419-VirtualBox:~/Desktop/lab3_output/exp2/5m/output$ ls -ltr
total 278340
-rw-r--r-- 1 cpre419 cpre419        0 Mar  6 19:52 _SUCCESS
-rw-r--r-- 1 cpre419 cpre419 28341426 Mar  6 19:52 part-r-00000
-rw-r--r-- 1 cpre419 cpre419 28790871 Mar  6 19:52 part-r-00001
-rw-r--r-- 1 cpre419 cpre419 28466256 Mar  6 19:52 part-r-00002
-rw-r--r-- 1 cpre419 cpre419 28372776 Mar  6 19:52 part-r-00003
-rw-r--r-- 1 cpre419 cpre419 28013619 Mar  6 19:52 part-r-00004
-rw-r--r-- 1 cpre419 cpre419 28835673 Mar  6 19:52 part-r-00005
-rw-r--r-- 1 cpre419 cpre419 29221449 Mar  6 19:52 part-r-00006
-rw-r--r-- 1 cpre419 cpre419 28581852 Mar  6 19:52 part-r-00007
-rw-r--r-- 1 cpre419 cpre419 27946701 Mar  6 19:52 part-r-00008
-rw-r--r-- 1 cpre419 cpre419 28429377 Mar  6 19:52 part-r-00009
```

As shown in the above figure, each part has approximately 1/10th of the data. This was done by considering 50000 samples for input sampling for 5m samples. This is just 1% not even 10% of the data and yet it does really well!!

First and Last 5 lines for custom parishioner on input-5m

cpre419@cpre419-VirtualBox:~/Desktop/lab3_output/exp2/5m/output$ ./print.sh
0
0001MeE7B7JPCWj Hslz3NRPzxugmQlH9s1UW1ku3jPc79T73HdtFxGJ
0001lcFyJCuKl4l vRlpXC3AuprUAMabMLgsIeSZxURRHKS0mi9VprSX
00023bbrDdDCyzn Un47nRNdoCLyDoh9irbhbRDuxlRgxB4HctA7Wubd
0006rvR9W2eYJL5 87fUZNdKD6C12lFoDagEcrJxZXce4sPTn4X57Qas
0008P8NCnnLVhJh qmRAduMXpSRodW8PCiPDtCqc5oJqLCuJaix9YFLp

63pVFg6M0SZeMj0 LdFrJXtkYbeBKjp2l5pz3AjSj3EGPJOtCtsLFnSV
63pVRUZmuzsniSF 7PKv4s3eSuDu7WJQsNQoeTio7EhHNWJNc9EXe4Mh
63pWJBOVN3MrtHi WN8yxHbbgiZyKKnWz2BlPeneOAoltB0TPiWhtlC6
63pZASnOh6MR3QV AU6x3FhkVRXT1pcrWOgeiVak32FLNl7iMlfPWCHn
63pc4fyYdKekJmF voLXIsKVHHeDaacNLm4ICZmpY349WmtOB4bZMfL7

1
63pmPxM5Jpg1mfL cLSVLlEu76MGUNIoP8msy47BPxslTalR8rVEIQhB
63ppIU5S15MyhAC tdlX3DzgExZ6XnHOAf4kWHD9t242stO21L66B8N3
63pre8CmnmduZ3g Gl0hgNyZOtRdpLTNlXlM5ESABQaG2uQRdjg5LM3V
63q0rsbVCPNPZSG XEcnO1k75Jfsj4m7AlVtx9PdYsRpt0LdFfU6mEqp
63q4rjQE0TQc0Ui FnEQ68jSPMYn8ST8lOtFzR0oObxpQ3J8MWqJ7vRA

CELZpcmTzDaCjmE 9QPXqUyYylljSOkUkLkVkRUYHjetKhVyAEcpn1Hz
CELaTPRpLQf0lgN ZmhuS6iAWkBSMWSceL2P35JEu8Jt9MQl3jEOZ9fc

CELezpbKWrWzpNo 44eSuKRoboJ1EtZCzfRzC7cfrkcla0VxI5kgzZ6m
CELhRcKmbdao9jz EyjNRgy2olcNk5xZsykGKlKCZMvAoyK7yhvpGPs3
CELmsZ27c7Bp0DE ZlEHn3LXNHRKp6zVnuIcBcErYmftzGMvTCS4Tqdi

2
CELoLDsen7WEyEs FlHE1C7piz3VlnMqqH4KvntfQo0iUfoCWtm53Ipq
CELsMRImMz099zj aGgc1msLzrRea97gU9DseIHNslHnu9JcOR7UE66P
CELsYReSYCSNJLM RNjJm2Oby0Ou6CxCJoczh5GBBb9TvEqcNOLEOXOm
CELvIDJ7KhiutyU FZriWdO31xhJyNq9qZ2TBMWQppPoxU9TmXKtRa6L
CELyT1RKTlc3250 zkrPsNGmvXnqjl6lQXYAU8XnebT8zKhMJpYEXEZt

lJREmtthA8A53iC Tc687EKj0LHI9mCaPpeKF14K6DXkgCbWxtFxn8fA
lJRNBYIfEgjObaL EXgV8q5QDumF4CyiPq2E3GmfCTkKCqWzP65mLLeo
lJRO3nGL6UcZAfk Ea5cihrpQ5oxe6y4sWroaSJ8LnpXtusLOOXE3dIs
lJRP33UXNvNY7W0 PFKgr1HiNcFVF0D04vD4bCkjX69MXI9oVFV6R0xQ
lJRRrnW2MGyc1OR MIW7xlBQ5mzBdt3XMXqFUkju1cPslkK1gHgK9F2b

3
lJRSY5VI1krIk9e fH77oSDsqHPs3kTyETQ1Do0mbLsBGlVuSCoI7YkF
lJRSYxaAxTjO9cc IqjxHIPYyyuP5SHlmtpBP7lB5vDTME5Eg31NZeUf
lJRURAdB1Vt2NEg pSXcUtYgEOu2tHxjMPSDgcjc3qlU7T4zE57o4ujh
lJRXXCk9LvxMgeN xXseRZCzNXvCysdqae3xkuWakPdUpeVmaMyj9iyg
lJRXxC4ZxJDXLju UO9xBp2oeFYXxmvF6pbjsINGNkNCCCExpafad7nt

OO8AFRj96OVcf85 osfLqoQMHJ152ujjTA3ncoa63qqLIr4t1GMR0zDO
OO8F0y9YoU3eikO oTI8T26ZXlCbcuUFm1CqIL7S9d3BiahOyuPSqr3q
OO8IVJoI8MZoJnQ szKKYq4oGxGZ5pjdHEq4eGZm2HjLdPzoHX0R6ejv
OO8JxeZ9XJ6GBtJ xosLZITcpD1kbiTuaqO5hSYD40h6JiFqqOnAC6FE
OO8Ka7zJ7mj1fmx e3cVzD7b5zcEs6Y7mCo4TuEXQrOOh9pXLdhY9UqX

4
OO8KaHV3cD4vYWG 5xYKNfqlOtfz225KitplzkHNDlFDN3J6zmIE9Yxs
OO8KqZ7hl2xW07N HGaBnQWLKtKjhvIGckYCffEZmaoN7lfBRZ98xx3b
OO8U8eRxVsHv87F qHr8M0RRfvVHhHj6RguRuTKi44A9NPL7LcOqeQUN
OO8WP3h9nN7Sguy sqnaeVqy0eBNkEvsA1P3IOtm9rTSMUXt5EUvdR6S
OO8Y3n1SeylvEzy eQq3ahBXnRT9fpaxsQP8qZiBIlAyTHsyqJufGpcs

UOAPCyfauH7v76T QobrsoxbsyJ0qdpsCavxqAsnCWqtKayG7eiDisjt
UOAUE9OtClhoKBI WNp4fRiXD62glZfioXDmAVe1jmd4voiUmCzOvhFM
UOAYZLt19Pz6kCa bQ7kndfLVcZTOdCoSWVKzvr1eBqCdNI6Bcs95oKO
UOAemWZdaKMPDE2 guPiut8li42RcBkBbqetaDWE6AtDLK5BJerFLMCl
UOAetLJelf2sZ1R SChHCThdZU2yKjJDJYKGGIM1WOpL45OdVMexFJDE

5
UOAfPum4BvVF4Nr gDWnJxKNkXUgRIET92aWRxxiNufmUdYVSRa2qA2c
UOAtzT2c9bdRD5V WfTRy3VLFm6QcEhvUuVlvDkUni1t2iaCttW4sOFT
UOAz4HYbkmmMIGF q7Qvl3itZ3iZ9jdH2VDuKJODA56ZBrEn0EoHJOVj
UOAz5vXTzC74i7T 9Fj46Yz3D5bkfg7WEsCgNFBjzlX7eYsjCJ72V0vH
UOAzl2EP9Bsl4yC iJLCEO2PMMAnFRn5hSnEBjWcIP0O9dFLR4LCkyBu

aXokdH8jFMqU3OC SrHZybKC9Qszeg4KahMdDJUHveOP1TLQQO09M1e9
aXot12tKF7gUn5b 2ZCbXqEftOlGgDhxDIYzZF2v9MU8hnWuh5H1a8yp
aXoucZEJQX63IIx Ws9Rc58ruU6Uu2WevPIJ0Cr8U3xoCTYpamhlgBUM
aXoy35dUNlceFnT iLT5KkO3US77IUVURdhnXMvuKVe2Ed5hVTzXqf2d
aXp0v0uXE0UEOFN 6YSLVUGxErRQORjVTHubNgWeRuiarSJvCMq5zgQj

6
aXp4nfgtyocy46p bmXC5pbN1GQcllDiuEGXbi22LJxK8bInjryVjfdS
aXp5mMFT84sYT0u T47SeJAj9z0N8Ru1eibhvE89NYA9n9O403tc380g
aXp6ctT2PdBZ7dS J3dQcTDrlx4MAP1lPdqMyuDLReFyJAMOUG33U0p6
aXp9S3LNPAOObkR fBAQJByr1obFd55zfmBKJqZ2eBk8ta2bicvCF2WX
aXpAoJllMoaRlzx Yfm5UXR1uafTbJNY5HG3nkEu2HzFM24slfmUriNx

gn2lIkIYeUnYFkq WcdccWMf3yVjJjCslgFkWhlnL9GiDbymVbTRVXDj
gn2mB7gCRVm38ur gQ0nPEfL4UQgDGe808Rg9Yz015UsDDRVZrK2RtVt
gn2qLaaqAhS29Ph OzfTRh1IyAsxVifqq2liG7gVkx31id8caac6IASa
gn2rITiIfhmvl70 x3PgqFPPt0mxCF3CPY9o6oQWDLJxIZt8y9gtqdqd
gn2s5Y83Qv2L0Jc xiFa1A8idQNiyPABd1hl3PSrJhBqxZSPX6LUcSTb

7
gn2sTsH2ldbjp9O vZkOT86NoCUdEYag1dILKd623N8xtyfMfY56hIDk
gn2ukS0WgCZlM2a iWahH7ngYRd61d3jB9nrXtk5gglWDncNJK1PG51L
gn2uyrnalpYzzLq EXc1JKAG85itombTWBzXmYjKJbGKFchj0dnve5EF
gn308UaZKb7kHKv 0XNeukfLLe8bBAQZMevfALbLxjxLE3EOGhXuferm

gn32OAkiBoq1LHJ zhyy4Ggu7paoa1SKmmexkGhmWHITHBqKkTYal1QU

mvN2euixkI9LjQO sHzbYqOGKr7OajmWPB8lGaWiOk5fl2DOlANLmBi7
mvN3ftmaI59bcVG eYg6jZ40jFpZJ4VyfZ97DyODgeR0zmH00DhhhBDs
mvN3iMtZ4crsoUX xr3m4VyPy2zChEPcsDSj380xORxN7a99P0FSMmO7
mvN8Ax2uHD9qtEe SseT3rS7H5QqH7qb5emonJktGCCep0enbbxoOWBy
mvN9mt6UAVYWHAk NOMaJ6loA5mMFKVK9gDXnsyyiyk4XnbixkJxbA6m

8
mvNEnEghCUEADO6 T9t32olNhlqOjAhcHAEe5WMF2JZaZ4L3NBeyF37C
mvNF9rnhIMt9buq zeoInu0BmYjtkZJdESqkEtUZpHLqjvNgYH1GQIte
mvNFTGNyU4d3FqW IGbRKW7UndFWhj5SLzcLIXQ6dU3TCRbsaHnX0ZN8
mvNGZ7BCdAK9QgV 4I0fFqrJGBIxkGF8iOzBR3fgp9PbGfMqGfpWhz0f
mvNHlm0BUy4Snrh moJgQREmJxvqqL4aC1stIqdWPOCMEJKQ31coQWLM

suWuEPARi3TU6xP RfcHZucB64IaLWVVrhbybmsFV9NEQAL38E7sP7o4
suWv3KDkl64TsTA pHfavzpNgN0U1y0FY176H86qEE0223nKSc50LbB3
suWytESX1eRjYae cQW3Gr4bck4KXzXOFb4HRvlj0Fp3xd5zFzTIFx4l
suWzDJBpdMJv6vy bvJUc68nIoiDcCL625gfOBzsdoPTKXAm2XpY2KQN
suX0eFqhBI8sOOr sr68Vaif2gIP1T3oxzbOtLDFN6Ol0mI5jkTM2kie

9
suX7qL1spb4XEaH q6q8DMmMiQjBnrf61Vp1TaIVRQIJTPxVQEgMWZfE
suX9crKYPxI2yBy Ukcr8lBZ7IeNjomPxJphhr7ifdk8RkdCrgMBpb8B
suXCMfkiFGgUU34 4MPtDM53AAydi8H1CXizG8pJXUKUNPv3E4O9USRQ
suXCjm9dxZXiAl0 HKphEXpWO3fKZ4bFV6iXNR5P9iur7PFEDdbHRz0V
suXELCKfeL4bSdq FPvn6vUUAxigxcequYQNxZEzyaV9aQDx2dvvYQDM

zzzlsNEhLA6XXr4 358YpGifuI8Wvb0f917CLkUTmcIMXn9UFpWFAiNm
zzznxiU8di17hBk zMz4cONqHlhp8PMnHaLtU7VB2pDlrS5Oo5kKYbip
zzzp2osLN7Didmk NA759V0qkWGK2Nfd8EvPc5FL9FgAIUbIX52JR4o3
zzzuoUL6N4qMAFu JzmJyyjISlKaSWPi5ObPcZoT9hEtHg3VDYQPTIco
zzzz0yxLW36DFE3 U32iJXn1rJyHPFOCvXEObCDsLT2pgv1IdMqBaeFI