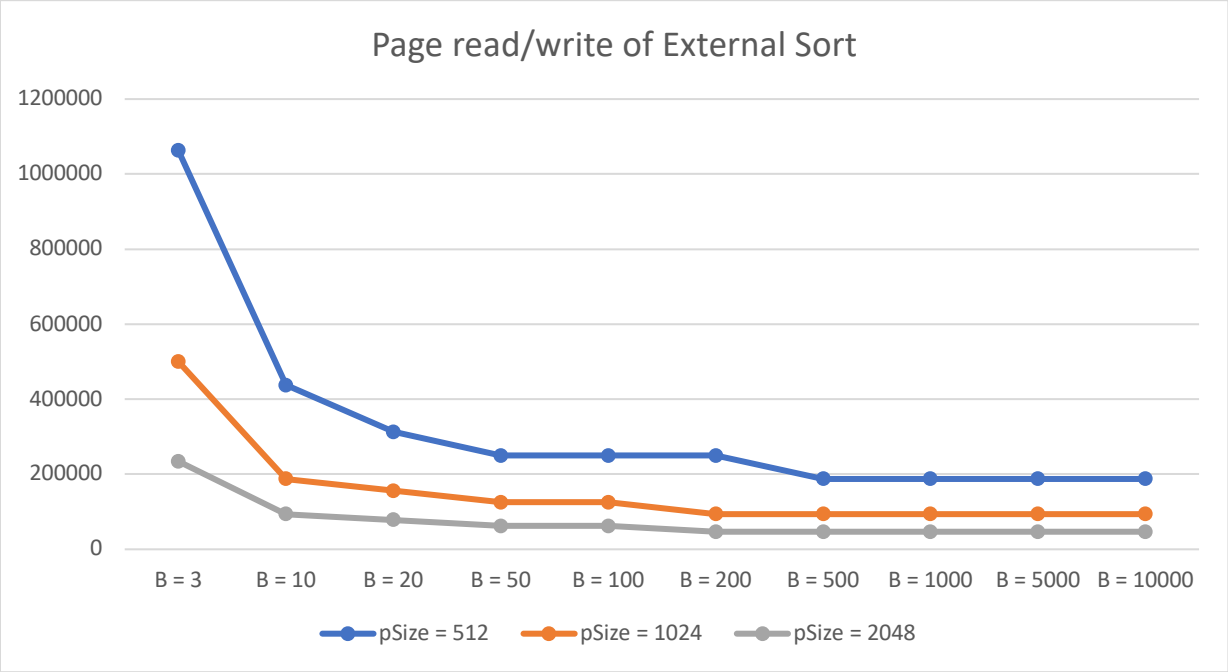
CSC443 Assignment 2 – Report

Part 1

In this part, we are required to write a program that perform external sort on a given database file. By adjust sorting parameters, like page size or number of buffers in used, we can see the performant difference in the number of pages read by the program.

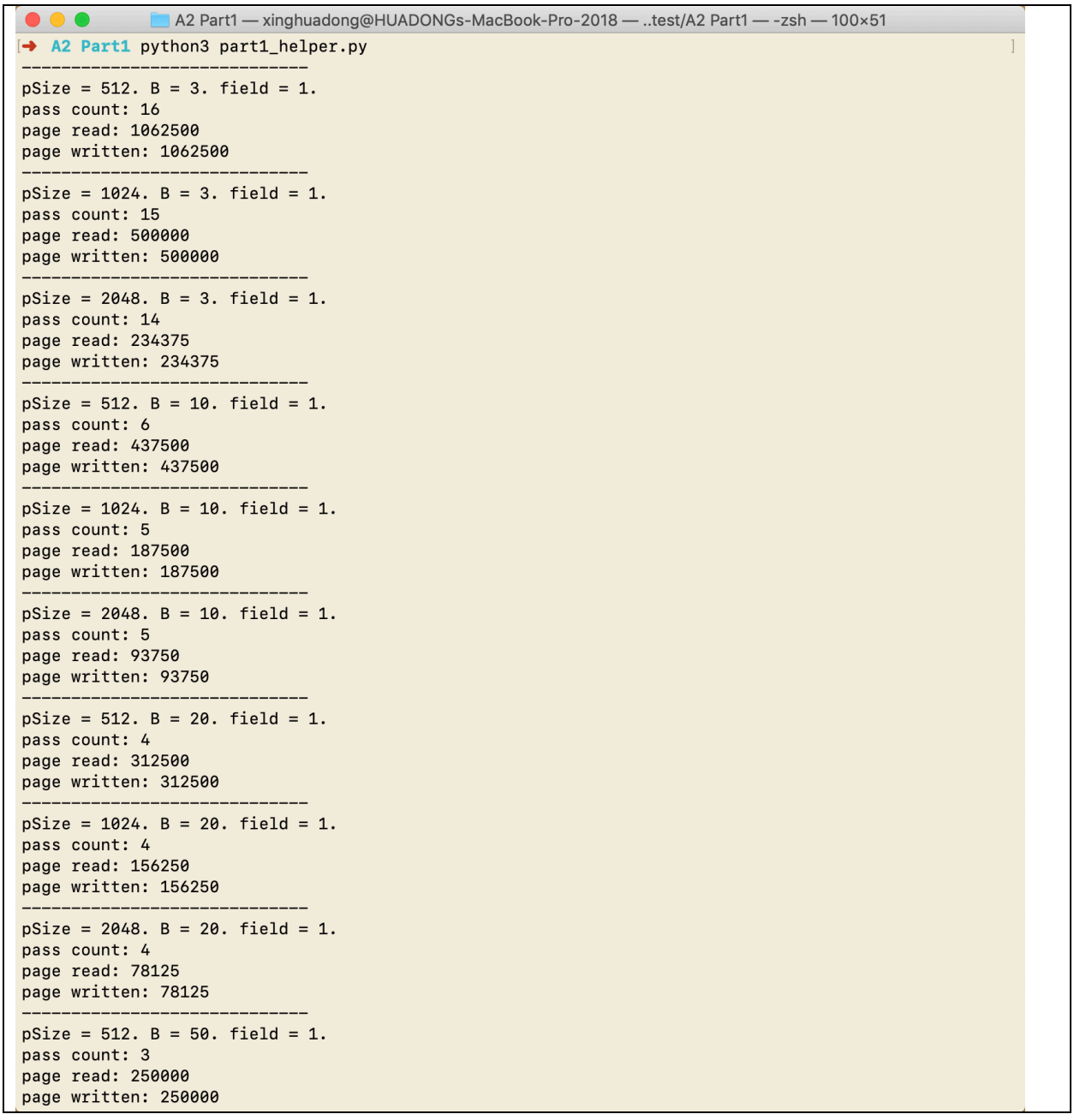
Here is the result summary. As we can see from buffer size = 3 to buffer size = 20, the number of pages read drop radically. Compare different page size, we can see the relation of page size and number of pages read is basically linear.

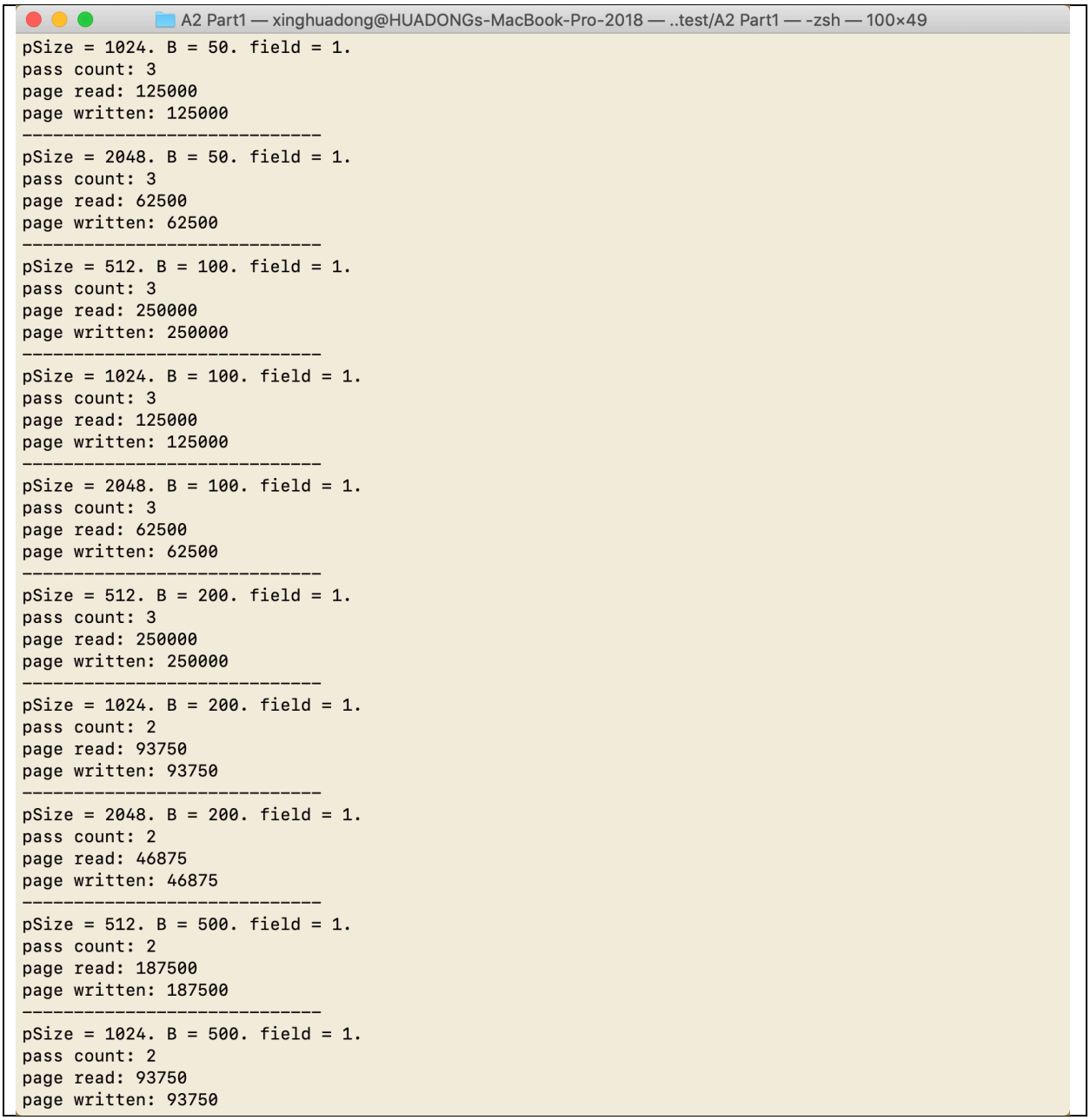


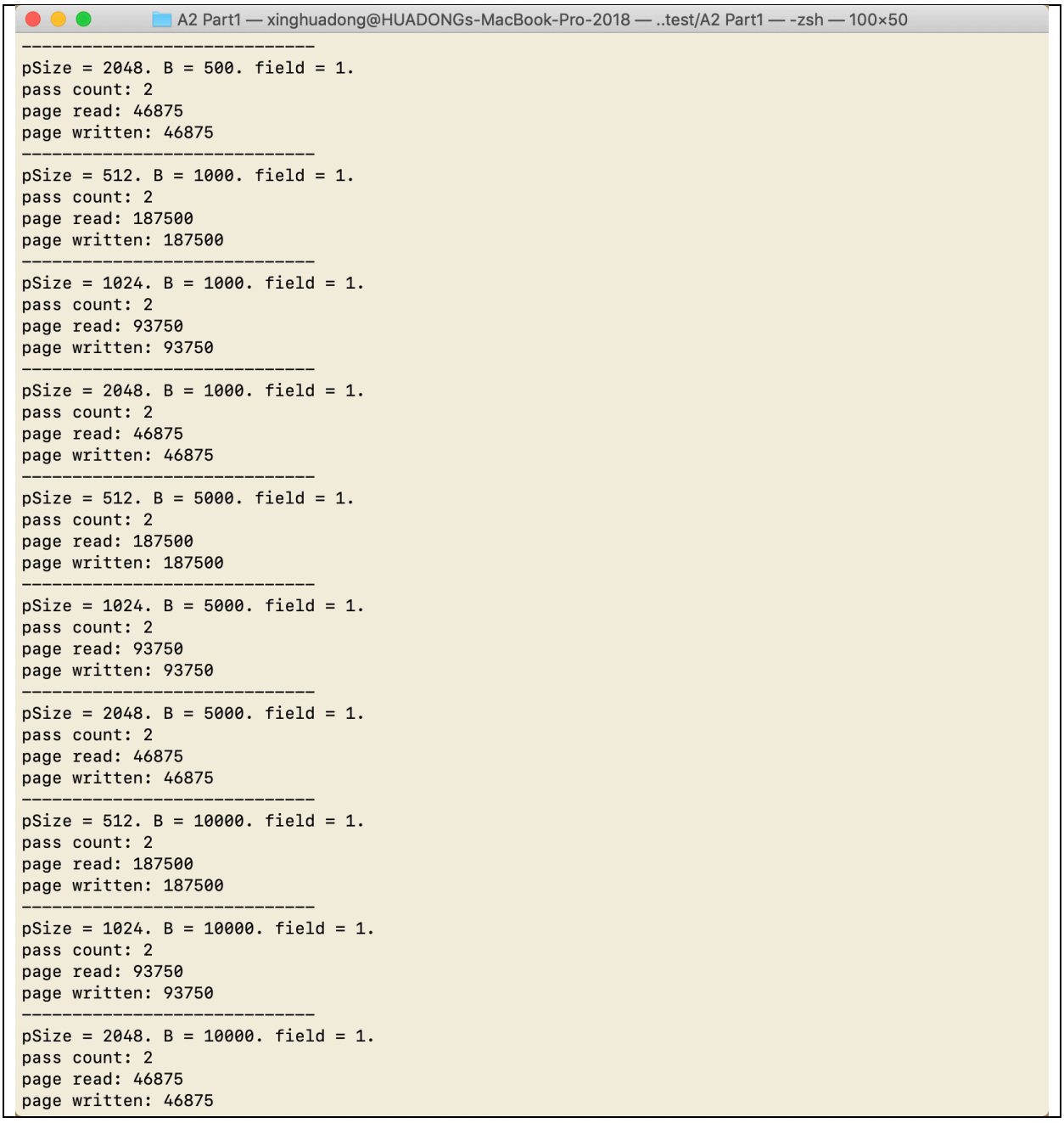
	pSize = 512	pSize = 1024	pSize = 2048
B = 3	1062500	500000	234375
B = 10	437500	187500	93750
B = 20	312500	156250	78125
B = 50	250000	125000	62500
B = 100	250000	125000	62500
B = 200	250000	93750	46875
B = 500	187500	93750	46875
B = 1000	187500	93750	46875
B = 5000	187500	93750	46875

B = 10000	187500	93750	46875

Here are the screenshots of the program output:




A2 Part1 — xinghuadong@HUADONGs-MacBook-Pro-2018 — ..test/A2 Part1 — -zsh — 100×49
pSize = 1024. B = 50. field = 1.
pass count: 3
page read: 125000
page written: 125000
----
pSize = 2048. B = 50. field = 1.
pass count: 3
page read: 62500
page written: 62500
----
pSize = 512. B = 100. field = 1.
pass count: 3
page read: 250000
page written: 250000
----
pSize = 1024. B = 100. field = 1.
pass count: 3
page read: 125000
page written: 125000
----
pSize = 2048. B = 100. field = 1.
pass count: 3
page read: 62500
page written: 62500
----
pSize = 512. B = 200. field = 1.
pass count: 3
page read: 250000
page written: 250000
----
pSize = 1024. B = 200. field = 1.
pass count: 2
page read: 93750
page written: 93750
----
pSize = 2048. B = 200. field = 1.
pass count: 2
page read: 46875
page written: 46875
----
pSize = 512. B = 500. field = 1.
pass count: 2
page read: 187500
page written: 187500
----
pSize = 1024. B = 500. field = 1.
pass count: 2
page read: 93750
page written: 93750


A2 Part1 — xinghuadong@HUADONGs-MacBook-Pro-2018 — ..test/A2 Part1 — -zsh — 100×50
----
pSize = 2048. B = 500. field = 1.
pass count: 2
page read: 46875
page written: 46875
----
pSize = 512. B = 1000. field = 1.
pass count: 2
page read: 187500
page written: 187500
----
pSize = 1024. B = 1000. field = 1.
pass count: 2
page read: 93750
page written: 93750
----
pSize = 2048. B = 1000. field = 1.
pass count: 2
page read: 46875
page written: 46875
----
pSize = 512. B = 5000. field = 1.
pass count: 2
page read: 187500
page written: 187500
----
pSize = 1024. B = 5000. field = 1.
pass count: 2
page read: 93750
page written: 93750
----
pSize = 2048. B = 5000. field = 1.
pass count: 2
page read: 46875
page written: 46875
----
pSize = 512. B = 10000. field = 1.
pass count: 2
page read: 187500
page written: 187500
----
pSize = 1024. B = 10000. field = 1.
pass count: 2
page read: 93750
page written: 93750
----
pSize = 2048. B = 10000. field = 1.
pass count: 2
page read: 46875
page written: 46875

Part 2

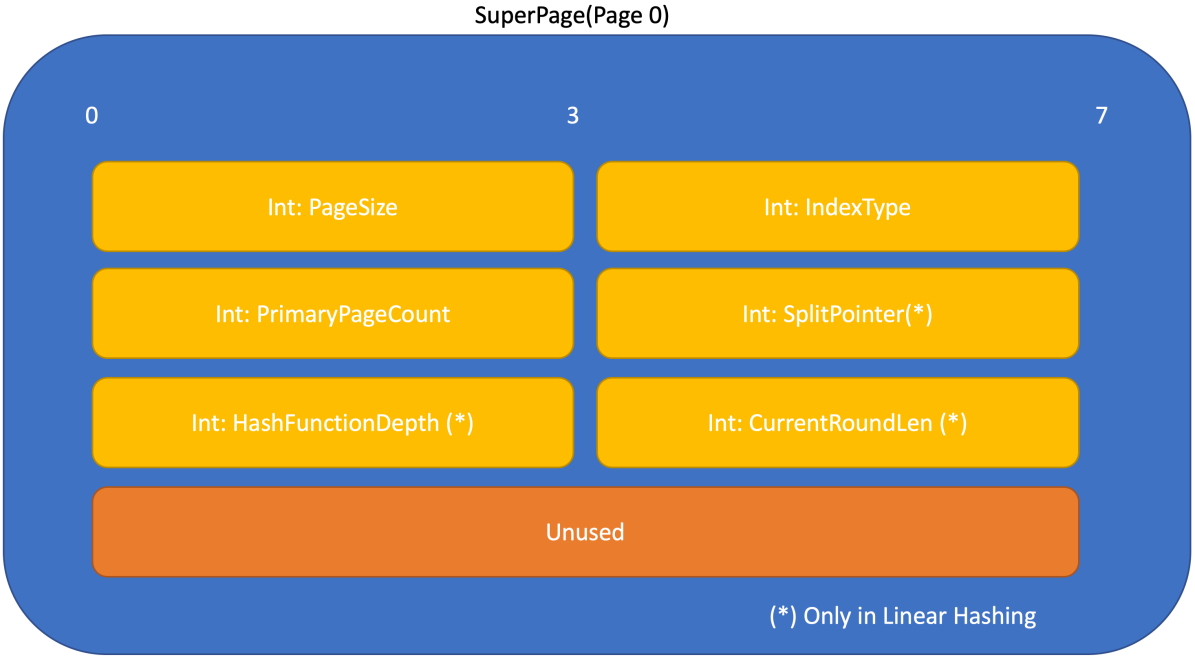
Index File Format

I use page to organize index file. There are three types of pages are in used in my design: Super Page, Reg/Overflow Page, Directory Page*(Only exist in Extendible Hashing).

Super Page

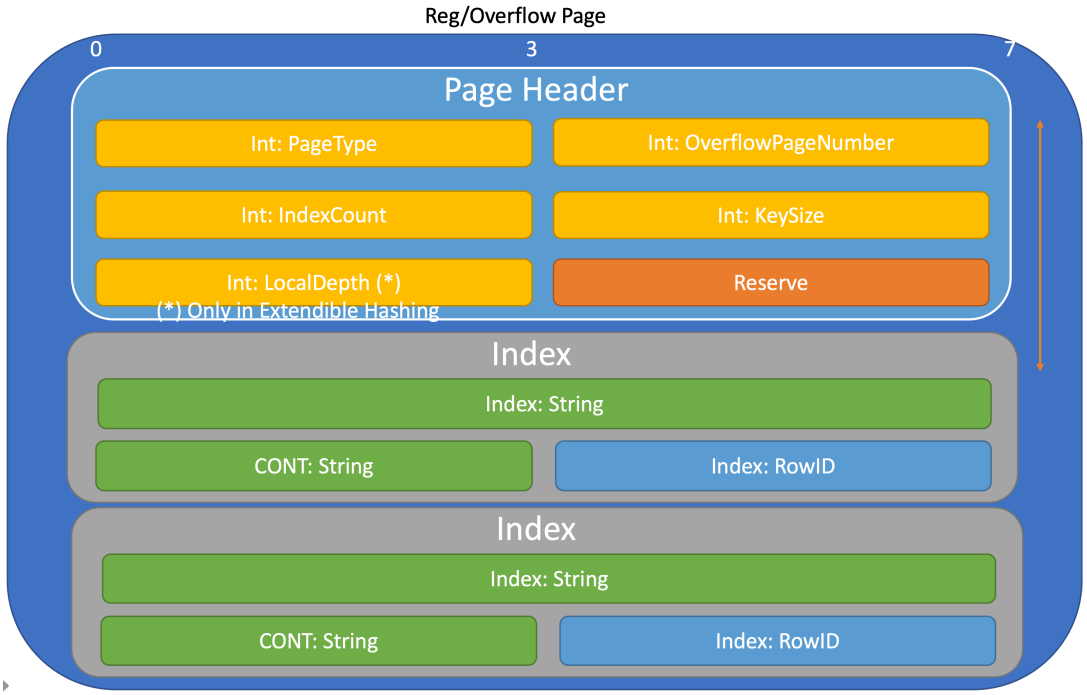
Super Page is always the first page (Page 0) of the Index file. It holds the most basic configuration of the Index file.
The first 4 bytes tells the page size of the Index file.
The second 4 bytes tells the Index type.
The third 4 bytes tells the number of primary pages in the Index file.
The fourth to sixth 4 bytes are only for Linear Hashing:

The fourth 4 bytes tells the position of Split Pointer.
The fifth 4 bytes tells the iteration of hash function.
The sixth 4 bytes tells the current round length.



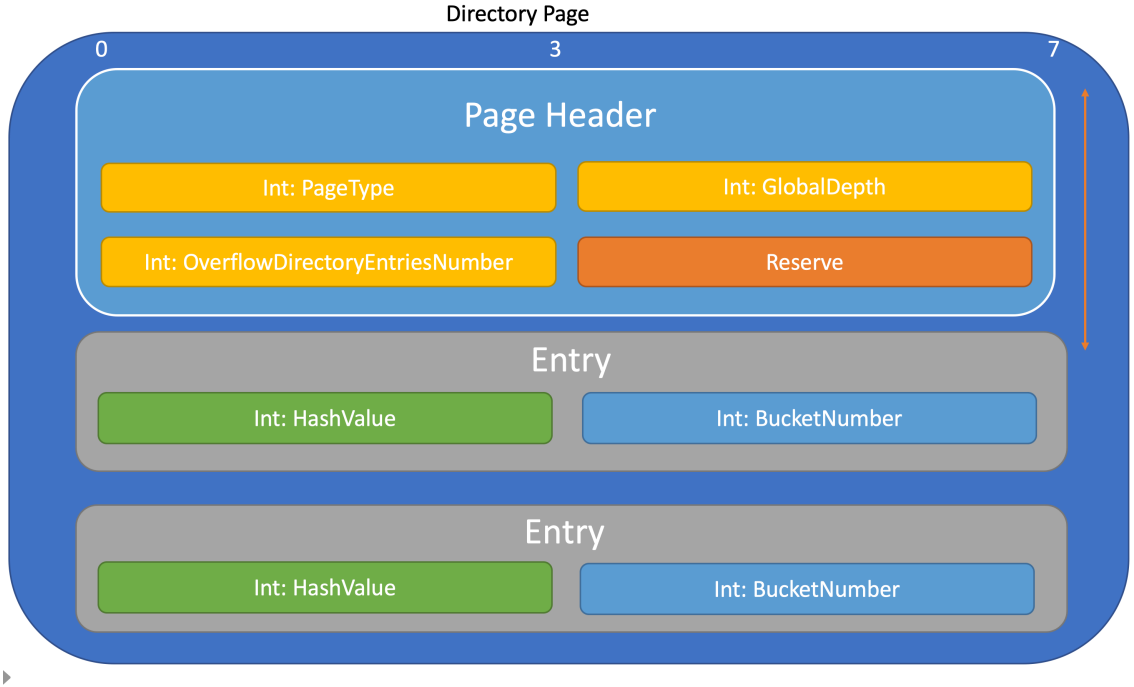
Reg/Overflow Page

Reg/Overflow page is the page type that holds Index data. The first 16 bytes (24 bytes for Extendible Hashing) is the page header. The header should hold: page type, the page number of the overflow page, the number of indices in this page, the size of key, and for extendible hashing we also keep track of Local Depth.
The indices data is following the Header. The first part is key, the length of key is specified by the header, the second part is a 4 bytes INT. So, each Index length is length of key + 4. In our part 2, we are creating index on First Name, so each Index length is 16 bytes. If page size is 1024, then each page will be able to holds 63 indices.

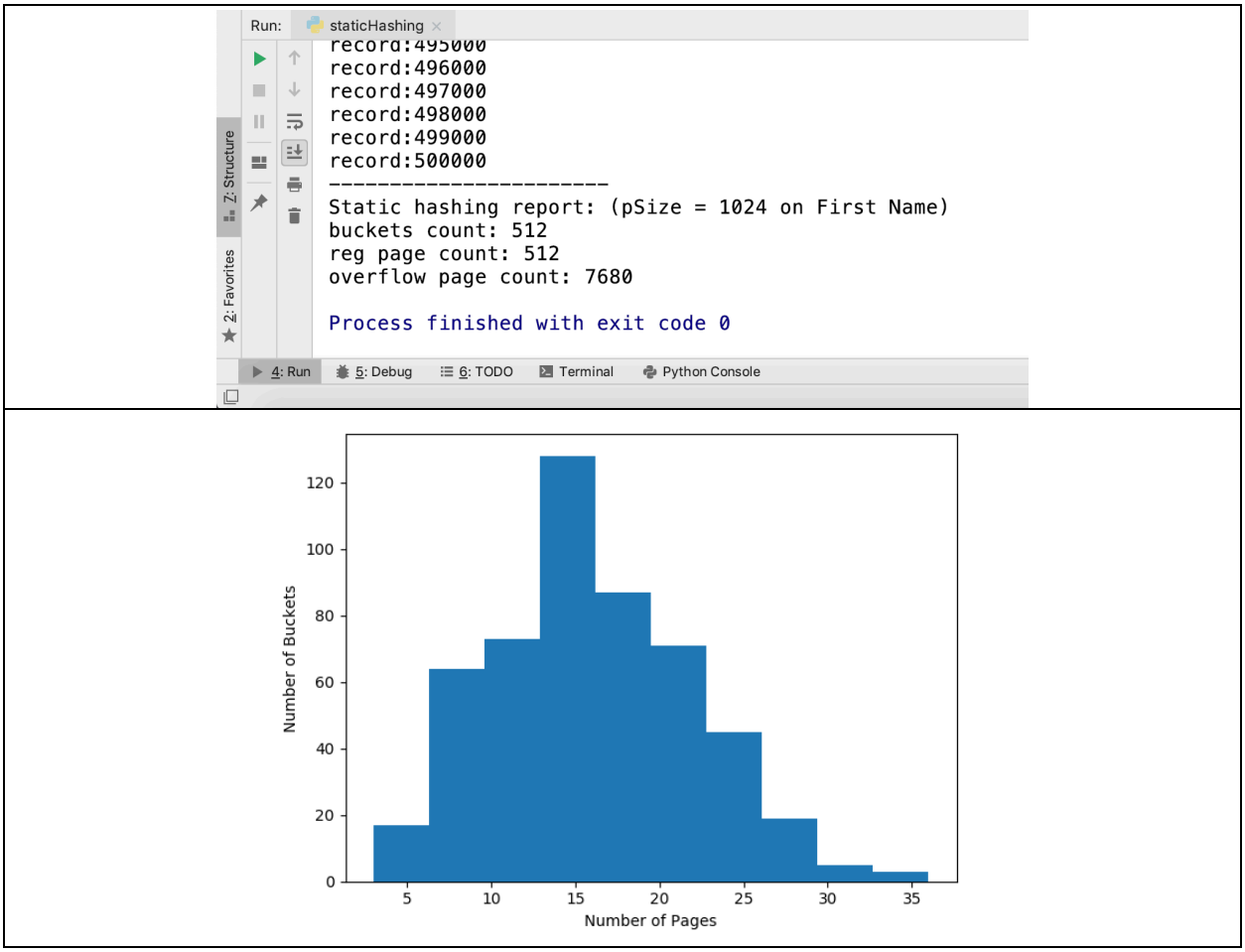


Directory Page

Directory Page is designed for Extendible Hashing. It is responsible for holding the directory info of Extendible Hashing including: Global Depth, Pointer to the overflow directory page and the entry for Bucket.



Static Hashing



As we can see, the number of pages per bucket is center at 15 pages. Because this is static hashing, overflow is allowing, so at the end of the run, all bucket has more than one page.

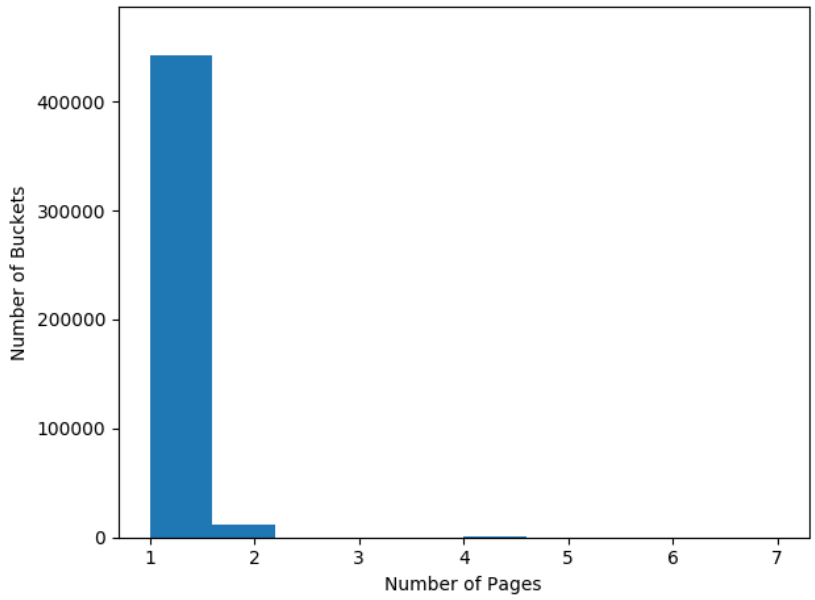
Although, overflow page may slow the searching process, but the index building speed is fastest amount those three hashing methods in this experiment.

Extendible Hashing

```
record:450000
record:460000
record:470000
record:480000
record:490000
record:500000
------------------------
Entendible hashing report: (pSize = 1024 on First Name)
buckets count: 466172
reg page count: 466172
overflow page count: 4813

Process finished with exit code 0
```

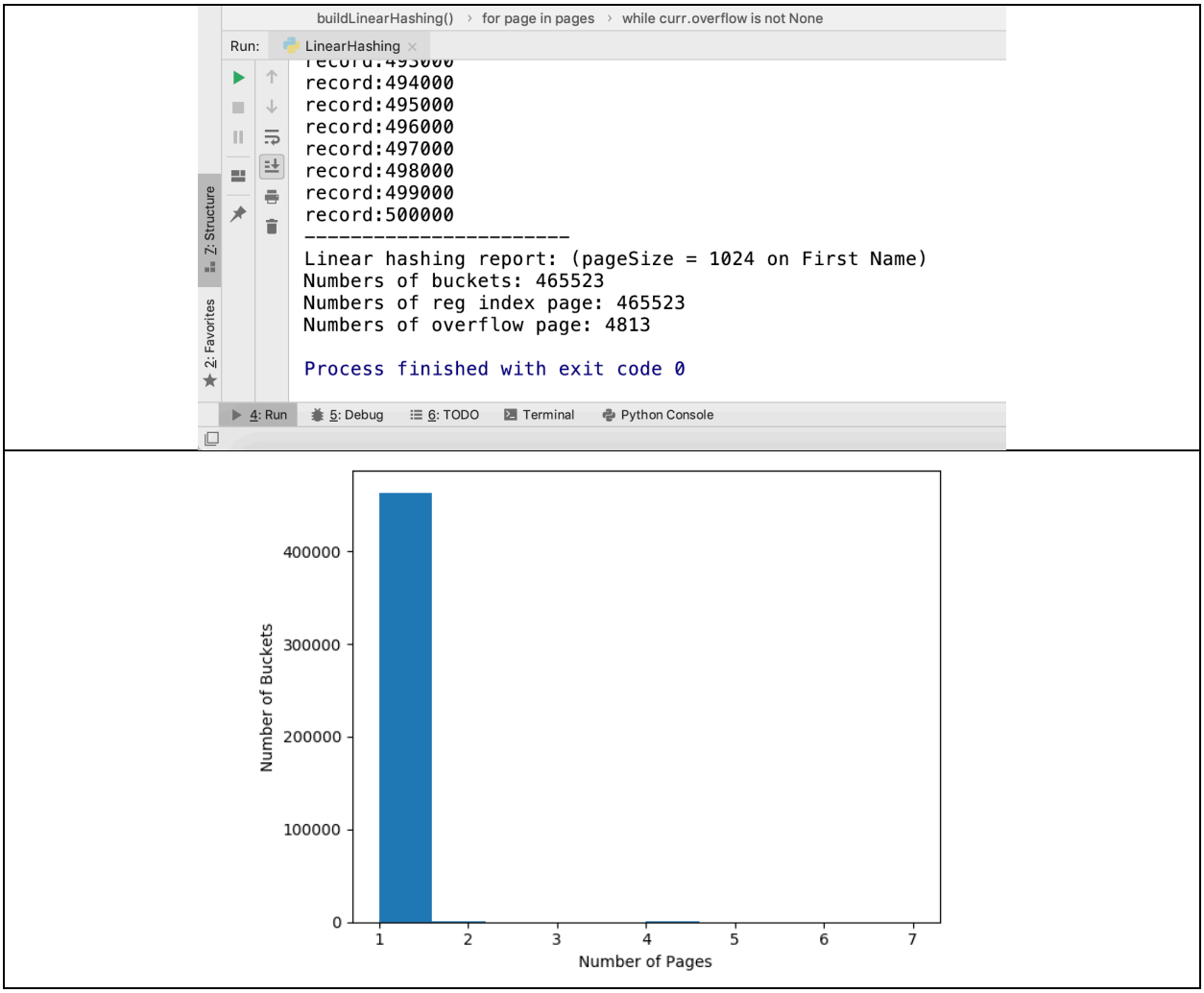
4: Run 5: Debug 6: TODO Terminal Python Console

PEP 8: blank line at end of file



As we can see, almost all buckets are having only one page and few of them have more than one.

Linear Hashing



We can see almost all buckets are having only one page and few of them have more than one. Since, this is Linear Hashing, it will try to avoid overflow as much as possible. Because there are many people have the same first name (can't be split by hashing), so that is why there are overflow page.

Part 3

In this part, we will perform a search (FirstName = 'Nona') in the database and use the index file that we generate in Part 2 to speed up the search (not writing to file in the part). During the experiment, the searching is extremely fast, the result show up immediately as we start running the program.

Search with Static Hashing Index

```
/usr/local/bin/python3.7 "/Users/xinghuadong/OneDrive/4th Year/443/A2/Part 3/read_index.py" names.db name_static.idx 0 Nona
FirstName   ,LastName      ,Email
Nona□□□□□□□□,Vizcaino□□□□□□,nona.vizcaino@gmail.com□□□□□□□□□□□□□□□□
Nona□□□□□□□□,Sipp□□□□□□□□□□,nona.sipp@gmail.com□□□□□□□□□□□□□□□□□□□□
Nona□□□□□□□□,Hirth□□□□□□□□□,nona.hirth@verizon.net□□□□□□□□□□□□□□□□
Nona□□□□□□□□,Mapp□□□□□□□□□□,nona.mapp@aol.com□□□□□□□□□□□□□□□□□□□□
Nona□□□□□□□□,Neil□□□□□□□□□□,nona.neil@msn.com□□□□□□□□□□□□□□□□□□□□
Nona□□□□□□□□,Quackenbush□□□,nona.quackenbush@charter.net□□□□□□□□□□
Nona□□□□□□□□,Preston□□□□□□□,nona.preston@aol.com□□□□□□□□□□□□□□□□□□
Nona□□□□□□□□,Mcclurg□□□□□□□,nona.mcclurg@rediffmail.com□□□□□□□□□□□
Nona□□□□□□□□,Peek□□□□□□□□□□,nona.peek@shell.com□□□□□□□□□□□□□□□□□□
Nona□□□□□□□□,Gilmartin□□□□□,nona.gilmartin@gmail.com□□□□□□□□□□□□□□
Nona□□□□□□□□,Vizcaino□□□□□□,nona.vizcaino@gmail.com□□□□□□□□□□□□□□□□
Nona□□□□□□□□,Snook□□□□□□□□□,nona.snook@hotmail.com□□□□□□□□□□□□□□□□
Nona□□□□□□□□,Kluesner□□□□□□,nona.kluesner@hotmail.com□□□□□□□□□□□□□□
Nona□□□□□□□□,Skalski□□□□□□□,nona.skalski@gmail.com□□□□□□□□□□□□□□□□
Nona□□□□□□□□,Hulbert□□□□□□□,nona.hulbert@msn.com□□□□□□□□□□□□□□□□□□
Nona□□□□□□□□,Evan□□□□□□□□□□,nona.evan@yahoo.co.uk□□□□□□□□□□□□□□□□□
Nona□□□□□□□□,Lucht□□□□□□□□□,nona.lucht@yahoo.ca□□□□□□□□□□□□□□□□□□□
Nona□□□□□□□□,Rowlands□□□□□□,nona.rowlands@aol.com□□□□□□□□□□□□□□□□□
Nona□□□□□□□□,Sheehan□□□□□□□,nona.sheehan@hotmail.com□□□□□□□□□□□□□□
Nona□□□□□□□□,Enlow□□□□□□□□□,nona.enlow@microsoft.com□□□□□□□□□□□□□□
Nona□□□□□□□□,Lamas□□□□□□□□□,nona.lamas@apple.com□□□□□□□□□□□□□□□□□□
Nona□□□□□□□□,Cabe□□□□□□□□□□,nona.cabe@aol.com□□□□□□□□□□□□□□□□□□□□
Nona□□□□□□□□,Buentello□□□□□,nona.buentello@yahoo.com□□□□□□□□□□□□□□□
Nona□□□□□□□□,Mccloskey□□□□□,nona.mccloskey@aol.com□□□□□□□□□□□□□□□□
Nona□□□□□□□□,Cote□□□□□□□□□□,nona.cote@yahoo.co.in□□□□□□□□□□□□□□□□
Nona□□□□□□□□,Arias□□□□□□□□□,nona.arias@aol.com□□□□□□□□□□□□□□□□□□□
Nona□□□□□□□□,Burgoon□□□□□□□,nona.burgoon@aol.com□□□□□□□□□□□□□□□□□□
Nona□□□□□□□□,Achenbach□□□□□,nona.achenbach@rediffmail.com□□□□□□□□□□
Nona□□□□□□□□,Budd□□□□□□□□□□,nona.budd@gmail.com□□□□□□□□□□□□□□□□□□□
Nona□□□□□□□□,Woodworth□□□□□,nona.woodworth@gmail.com□□□□□□□□□□□□□□□
Nona□□□□□□□□,Lilly□□□□□□□□□,nona.lilly@yahoo.com□□□□□□□□□□□□□□□□□□
Nona□□□□□□□□,Edgar□□□□□□□□□,nona.edgar@yahoo.com□□□□□□□□□□□□□□□□□□
Nona□□□□□□□□,Forest□□□□□□□□,nona.forest@exxonmobil.com□□□□□□□□□□□□□
Nona□□□□□□□□,Cyr□□□□□□□□□□□,nona.cyr@gmail.com□□□□□□□□□□□□□□□□□□□
Nona□□□□□□□□,Catchings□□□□□,nona.catchings@bellsouth.net□□□□□□□□□□□
Nona□□□□□□□□,Lansberry□□□□□,nona.lansberry@yahoo.com□□□□□□□□□□□□□□□
Nona□□□□□□□□,Bloomberg□□□□□,nona.bloomberg@gmail.com□□□□□□□□□□□□□□□
Nona□□□□□□□□,Ventimiglia□□□,nona.ventimiglia@gmail.com□□□□□□□□□□□□□
Nona□□□□□□□□,Belliveau□□□□□,nona.belliveau@gmail.com□□□□□□□□□□□□□□□
Total record found: 39 records

Summery:
Number of index page(primary + overflow) read: 10
Number of data(main db) page read: 39
```

Due to there are many overflow pages in static hashing, we can see it need 10 pages of index read to find all record.

Search with Extendible Hashing Index

```
/usr/local/bin/python3.7 "/Users/xinghuadong/OneDrive/4th Year/443/A2/Part 3/read_index.py" names.db name_extendible.idx 1 Nona
FirstName   ,LastName      ,Email
Nona□□□□□□□□,Vizcaino□□□□□□,nona.vizcaino@gmail.com□□□□□□□□□□□□□□□□
Nona□□□□□□□□,Sipp□□□□□□□□□,nona.sipp@gmail.com□□□□□□□□□□□□□□□□□□□□
Nona□□□□□□□□,Hirth□□□□□□□□,nona.hirth@verizon.net□□□□□□□□□□□□□□□□
Nona□□□□□□□□,Mapp□□□□□□□□,nona.mapp@aol.com□□□□□□□□□□□□□□□□□□□□□
Nona□□□□□□□□,Neil□□□□□□□□,nona.neil@msn.com□□□□□□□□□□□□□□□□□□□□□
Nona□□□□□□□□,Quackenbush□□,nona.quackenbush@charter.net□□□□□□□□□□
Nona□□□□□□□□,Preston□□□□□□,nona.preston@aol.com□□□□□□□□□□□□□□□□□
Nona□□□□□□□□,Mcclurg□□□□□□,nona.mcclurg@rediffmail.com□□□□□□□□□□
Nona□□□□□□□□,Peek□□□□□□□□,nona.peek@shell.com□□□□□□□□□□□□□□□□□□
Nona□□□□□□□□,Gilmartin□□□□□,nona.gilmartin@gmail.com□□□□□□□□□□□□
Nona□□□□□□□□,Vizcaino□□□□□□,nona.vizcaino@gmail.com□□□□□□□□□□□□□
Nona□□□□□□□□,Snook□□□□□□□□,nona.snook@hotmail.com□□□□□□□□□□□□□□
Nona□□□□□□□□,Kluesner□□□□□□,nona.kluesner@hotmail.com□□□□□□□□□□□
Nona□□□□□□□□,Skalski□□□□□□□,nona.skalski@gmail.com□□□□□□□□□□□□□□
Nona□□□□□□□□,Hulbert□□□□□□□,nona.hulbert@msn.com□□□□□□□□□□□□□□□
Nona□□□□□□□□,Evan□□□□□□□□□,nona.evan@yahoo.co.uk□□□□□□□□□□□□□□□
Nona□□□□□□□□,Lucht□□□□□□□□,nona.lucht@yahoo.ca□□□□□□□□□□□□□□□□□
Nona□□□□□□□□,Rowlands□□□□□□,nona.rowlands@aol.com□□□□□□□□□□□□□□□
Nona□□□□□□□□,Sheehan□□□□□□□,nona.sheehan@hotmail.com□□□□□□□□□□□□
Nona□□□□□□□□,Enlow□□□□□□□□,nona.enlow@microsoft.com□□□□□□□□□□□□
Nona□□□□□□□□,Lamas□□□□□□□□,nona.lamas@apple.com□□□□□□□□□□□□□□□□
Nona□□□□□□□□,Cabe□□□□□□□□□,nona.cabe@aol.com□□□□□□□□□□□□□□□□□□□
Nona□□□□□□□□,Buentello□□□□□,nona.buentello@yahoo.com□□□□□□□□□□□□
Nona□□□□□□□□,Mccloskey□□□□□,nona.mccloskey@aol.com□□□□□□□□□□□□□
Nona□□□□□□□□,Cote□□□□□□□□□,nona.cote@yahoo.co.in□□□□□□□□□□□□□□□
Nona□□□□□□□□,Arias□□□□□□□□,nona.arias@aol.com□□□□□□□□□□□□□□□□□□
Nona□□□□□□□□,Burgoon□□□□□□,nona.burgoon@aol.com□□□□□□□□□□□□□□□□
Nona□□□□□□□□,Achenbach□□□□□,nona.achenbach@rediffmail.com□□□□□□□□
Nona□□□□□□□□,Budd□□□□□□□□□,nona.budd@gmail.com□□□□□□□□□□□□□□□□□
Nona□□□□□□□□,Woodworth□□□□□,nona.woodworth@gmail.com□□□□□□□□□□□□
Nona□□□□□□□□,Lilly□□□□□□□□,nona.lilly@yahoo.com□□□□□□□□□□□□□□□□
Nona□□□□□□□□,Edgar□□□□□□□□,nona.edgar@yahoo.com□□□□□□□□□□□□□□□□
Nona□□□□□□□□,Forest□□□□□□□,nona.forest@exxonmobil.com□□□□□□□□□□□
Nona□□□□□□□□,Cyr□□□□□□□□□,nona.cyr@gmail.com□□□□□□□□□□□□□□□□□□□
Nona□□□□□□□□,Catchings□□□□□,nona.catchings@bellsouth.net□□□□□□□□□
Nona□□□□□□□□,Lansberry□□□□□,nona.lansberry@yahoo.com□□□□□□□□□□□□
Nona□□□□□□□□,Bloomberg□□□□□,nona.bloomberg@gmail.com□□□□□□□□□□□□
Nona□□□□□□□□,Ventimiglia□□□,nona.ventimiglia@gmail.com□□□□□□□□□□□
Nona□□□□□□□□,Belliveau□□□□□,nona.belliveau@gmail.com□□□□□□□□□□□□□
Total record found: 39 records

Summery:
Number of index page(primary + overflow) read: 1
Number of data(main db) page read: 39

Process finished with exit code 0
```

Since number of record that First Name = ‘Nona’ (39 indices) is less than the capacity of page to holds index (63 indices per page) as we discussed in Part 2, it makes sense that we only read one index page to find all records.

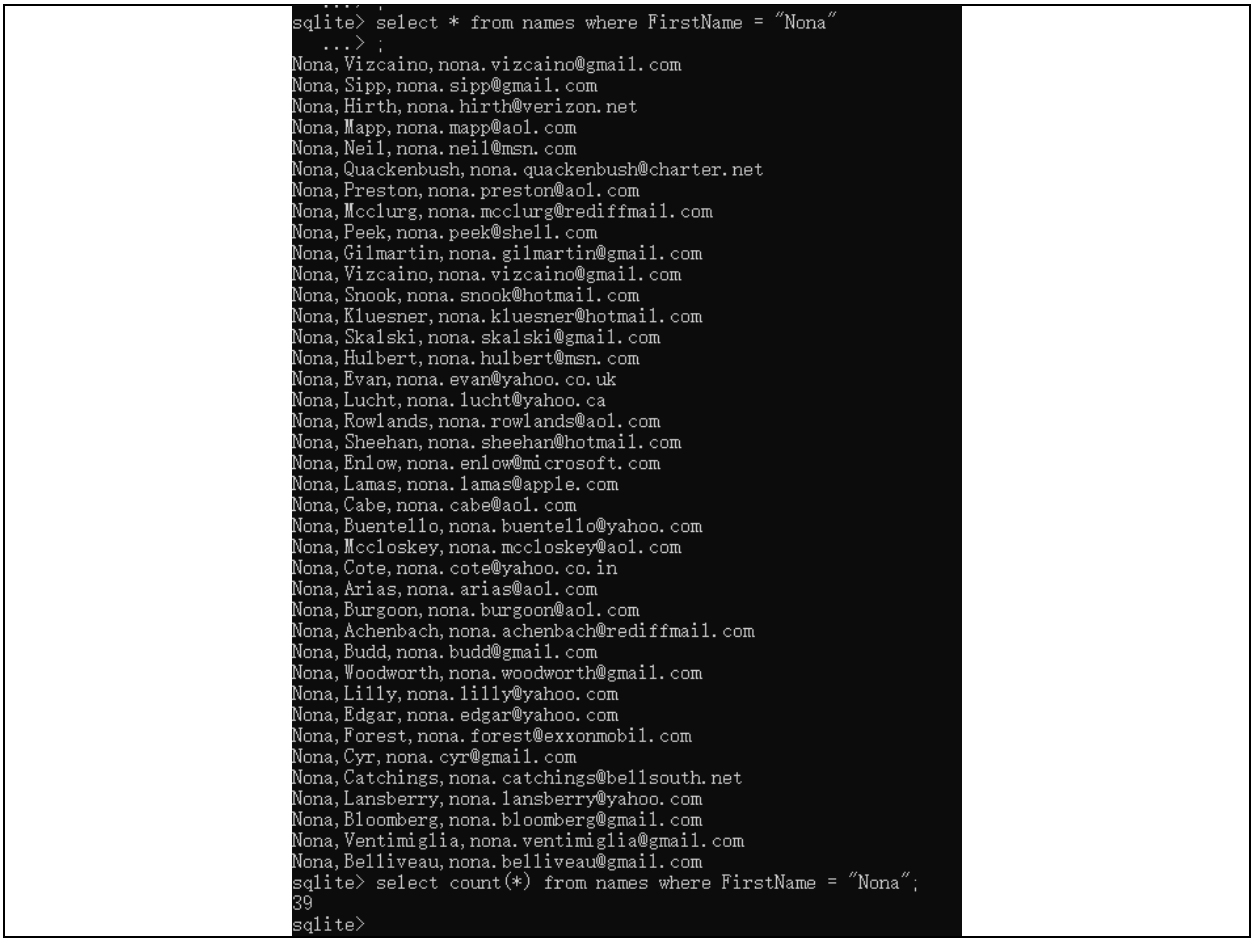
Search with Linear Hashing Index

```
/usr/local/bin/python3.7 "/Users/xinghuadong/OneDrive/4th Year/443/A2/Part 3/read_index.py" names.db name_linear.idx 2 Nona
FirstName   ,LastName      ,Email
Nona□□□□□□□□,Vizcaino□□□□□□,nona.vizcaino@gmail.com□□□□□□□□□□□□□□□□
Nona□□□□□□□□,Sipp□□□□□□□□□□,nona.sipp@gmail.com□□□□□□□□□□□□□□□□□□
Nona□□□□□□□□,Hirth□□□□□□□□□,nona.hirth@verizon.net□□□□□□□□□□□□□□□
Nona□□□□□□□□,Mapp□□□□□□□□□□,nona.mapp@aol.com□□□□□□□□□□□□□□□□□□□
Nona□□□□□□□□,Neil□□□□□□□□□□,nona.neil@msn.com□□□□□□□□□□□□□□□□□□□
Nona□□□□□□□□,Quackenbush□□□,nona.quackenbush@charter.net□□□□□□□□□□
Nona□□□□□□□□,Preston□□□□□□□,nona.preston@aol.com□□□□□□□□□□□□□□□□□
Nona□□□□□□□□,Mcclurg□□□□□□□,nona.mcclurg@rediffmail.com□□□□□□□□□□
Nona□□□□□□□□,Peek□□□□□□□□□□,nona.peek@shell.com□□□□□□□□□□□□□□□□□
Nona□□□□□□□□,Gilmartin□□□□□,nona.gilmartin@gmail.com□□□□□□□□□□□□□
Nona□□□□□□□□,Vizcaino□□□□□□,nona.vizcaino@gmail.com□□□□□□□□□□□□□□
Nona□□□□□□□□,Snook□□□□□□□□□,nona.snook@hotmail.com□□□□□□□□□□□□□□□
Nona□□□□□□□□,Kluesner□□□□□□,nona.kluesner@hotmail.com□□□□□□□□□□□□
Nona□□□□□□□□,Skalski□□□□□□□,nona.skalski@gmail.com□□□□□□□□□□□□□□□
Nona□□□□□□□□,Hulbert□□□□□□□,nona.hulbert@msn.com□□□□□□□□□□□□□□□□□
Nona□□□□□□□□,Evan□□□□□□□□□,nona.evan@yahoo.co.uk□□□□□□□□□□□□□□□□
Nona□□□□□□□□,Lucht□□□□□□□□□,nona.lucht@yahoo.ca□□□□□□□□□□□□□□□□□
Nona□□□□□□□□,Rowlands□□□□□□,nona.rowlands@aol.com□□□□□□□□□□□□□□□□
Nona□□□□□□□□,Sheehan□□□□□□□,nona.sheehan@hotmail.com□□□□□□□□□□□□□
Nona□□□□□□□□,Enlow□□□□□□□□□,nona.enlow@microsoft.com□□□□□□□□□□□□□
Nona□□□□□□□□,Lamas□□□□□□□□□,nona.lamas@apple.com□□□□□□□□□□□□□□□□□
Nona□□□□□□□□,Cabe□□□□□□□□□□,nona.cabe@aol.com□□□□□□□□□□□□□□□□□□□
Nona□□□□□□□□,Buentello□□□□□,nona.buentello@yahoo.com□□□□□□□□□□□□□□
Nona□□□□□□□□,Mccloskey□□□□□,nona.mccloskey@aol.com□□□□□□□□□□□□□□□
Nona□□□□□□□□,Cote□□□□□□□□□□,nona.cote@yahoo.co.in□□□□□□□□□□□□□□□
Nona□□□□□□□□,Arias□□□□□□□□□,nona.arias@aol.com□□□□□□□□□□□□□□□□□□
Nona□□□□□□□□,Burgoon□□□□□□□,nona.burgoon@aol.com□□□□□□□□□□□□□□□□
Nona□□□□□□□□,Achenbach□□□□□,nona.achenbach@rediffmail.com□□□□□□□□□
Nona□□□□□□□□,Budd□□□□□□□□□□,nona.budd@gmail.com□□□□□□□□□□□□□□□□□
Nona□□□□□□□□,Woodworth□□□□□,nona.woodworth@gmail.com□□□□□□□□□□□□□
Nona□□□□□□□□,Lilly□□□□□□□□□,nona.lilly@yahoo.com□□□□□□□□□□□□□□□□□
Nona□□□□□□□□,Edgar□□□□□□□□□,nona.edgar@yahoo.com□□□□□□□□□□□□□□□□
Nona□□□□□□□□,Forest□□□□□□□□,nona.forest@exxonmobil.com□□□□□□□□□□□
Nona□□□□□□□□,Cyr□□□□□□□□□□,nona.cyr@gmail.com□□□□□□□□□□□□□□□□□□
Nona□□□□□□□□,Catchings□□□□□,nona.catchings@bellsouth.net□□□□□□□□□□
Nona□□□□□□□□,Lansberry□□□□□,nona.lansberry@yahoo.com□□□□□□□□□□□□□
Nona□□□□□□□□,Bloomberg□□□□□,nona.bloomberg@gmail.com□□□□□□□□□□□□□
Nona□□□□□□□□,Ventimiglia□□□,nona.ventimiglia@gmail.com□□□□□□□□□□□
Nona□□□□□□□□,Belliveau□□□□□,nona.belliveau@gmail.com□□□□□□□□□□□□□
Total record found: 39 records

Summery:
Number of index page(primary + overflow) read: 1
Number of data(main db) page read: 39

Process finished with exit code 0
```

Same reason as Extendible Hashing, we only need one page to holds all the index of ‘Nona’ records.

Reference Result

In order to make sure the result given by the searching program is correct, I used an existed DBMS to verify the result from my program. I first import the original records (names.db) into SQLite and use SQL to search the matched record. As we can see, the result matches each other.



```
sqlite> select * from names where FirstName = "Nona"
   ...> ;
Nona,Vizcaino,nona.vizcaino@gmail.com
Nona,Sipp,nona.sipp@gmail.com
Nona,Hirth,nona.hirth@verizon.net
Nona,Mapp,nona.mapp@aol.com
Nona,Neil,nona.neil@msn.com
Nona,Quackenbush,nona.quackenbush@charter.net
Nona,Preston,nona.preston@aol.com
Nona,Mcclurg,nona.mcclurg@rediffmail.com
Nona,Peek,nona.peek@shell.com
Nona,Gilmartin,nona.gilmartin@gmail.com
Nona,Vizcaino,nona.vizcaino@gmail.com
Nona,Snook,nona.snook@hotmail.com
Nona,Kluesner,nona.kluesner@hotmail.com
Nona,Skalski,nona.skalski@gmail.com
Nona,Hulbert,nona.hulbert@msn.com
Nona,Evan,nona.evan@yahoo.co.uk
Nona,Lucht,nona.lucht@yahoo.ca
Nona,Rowlands,nona.rowlands@aol.com
Nona,Sheehan,nona.sheehan@hotmail.com
Nona,Enlow,nona.enlow@microsoft.com
Nona,Lamas,nona.lamas@apple.com
Nona,Cabe,nona.cabe@aol.com
Nona,Buentello,nona.buentello@yahoo.com
Nona,Mccloskey,nona.mccloskey@aol.com
Nona,Cote,nona.cote@yahoo.co.in
Nona,Arias,nona.arias@aol.com
Nona,Burgoon,nona.burgoon@aol.com
Nona,Achenbach,nona.achenbach@rediffmail.com
Nona,Budd,nona.budd@gmail.com
Nona,Woodworth,nona.woodworth@gmail.com
Nona,Lilly,nona.lilly@yahoo.com
Nona,Edgar,nona.edgar@yahoo.com
Nona,Forest,nona.forest@exxonmobil.com
Nona,Cyr,nona.cyr@gmail.com
Nona,Catchings,nona.catchings@bellsouth.net
Nona,Lansberry,nona.lansberry@yahoo.com
Nona,Bloomberg,nona.bloomberg@gmail.com
Nona,Ventimiglia,nona.ventimiglia@gmail.com
Nona,Belliveau,nona.belliveau@gmail.com
sqlite> select count(*) from names where FirstName = "Nona";
39
sqlite>
```