

Journal

Yitan Lou

S1996177

dabing930714@gmail.com

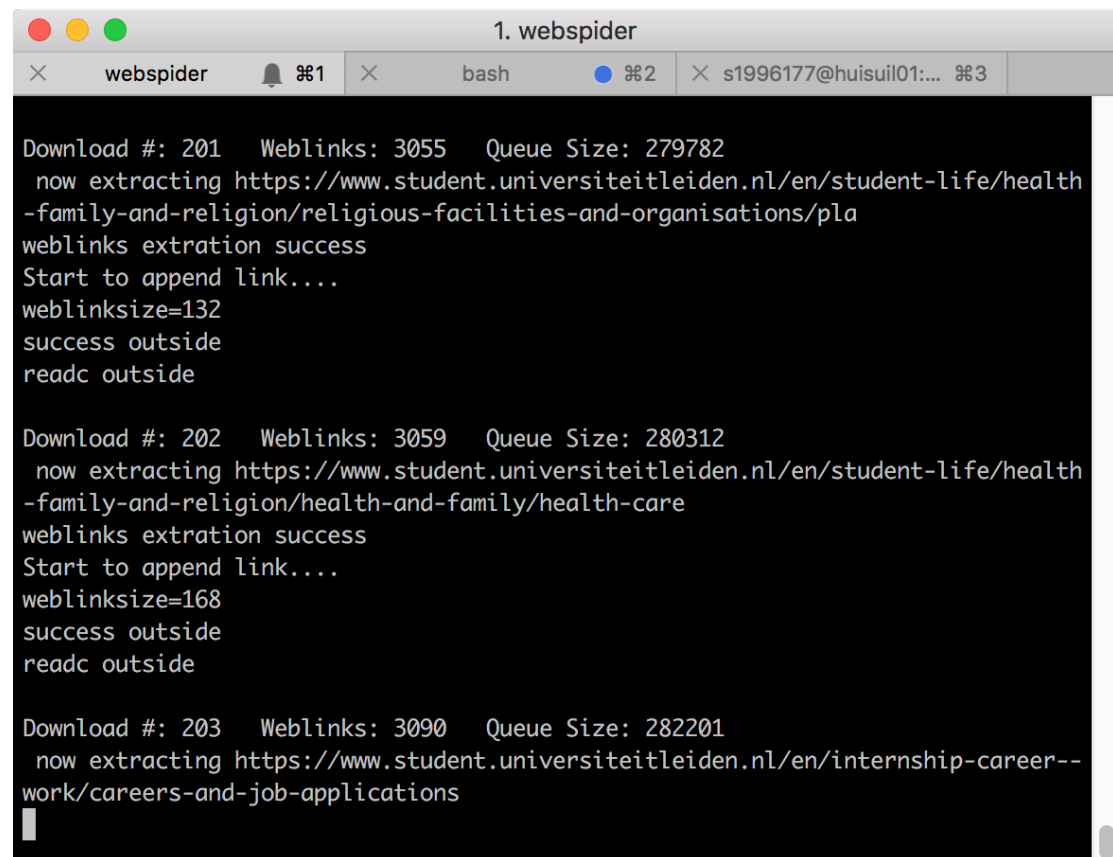
1) Duplicate Detection

in this section I choose medium difficulty (use linked list and binary tree to check for the duplicates.

With this method, every time the program must check , whether the current url is already in the queue or not. Only in case of the url has not been met before, the weblink will be stored into the queue and the tree.

Specific code are attached.(related code are mainly in function Appendlink2 and the data structure are defined in file BStree.h)

Screenshot:



```
1. webspider
× webspider  %1 × bash %2 × s1996177@huisil01:... %3

Download #: 201  Weblinks: 3055  Queue Size: 279782
now extracting https://www.student.universiteitleid.nl/en/student-life/health
-family-and-religion/religious-facilities-and-organisations/pla
weblinks extration success
Start to append link...
weblinksize=132
success outside
readc outside

Download #: 202  Weblinks: 3059  Queue Size: 280312
now extracting https://www.student.universiteitleid.nl/en/student-life/health
-family-and-religion/health-and-family/health-care
weblinks extration success
Start to append link...
weblinksize=168
success outside
readc outside

Download #: 203  Weblinks: 3090  Queue Size: 282201
now extracting https://www.student.universiteitleid.nl/en/internship-career--
work/careers-and-job-applications
```

In compare with the former program without duplicate detection, the size of weblinks shrinks.

2)

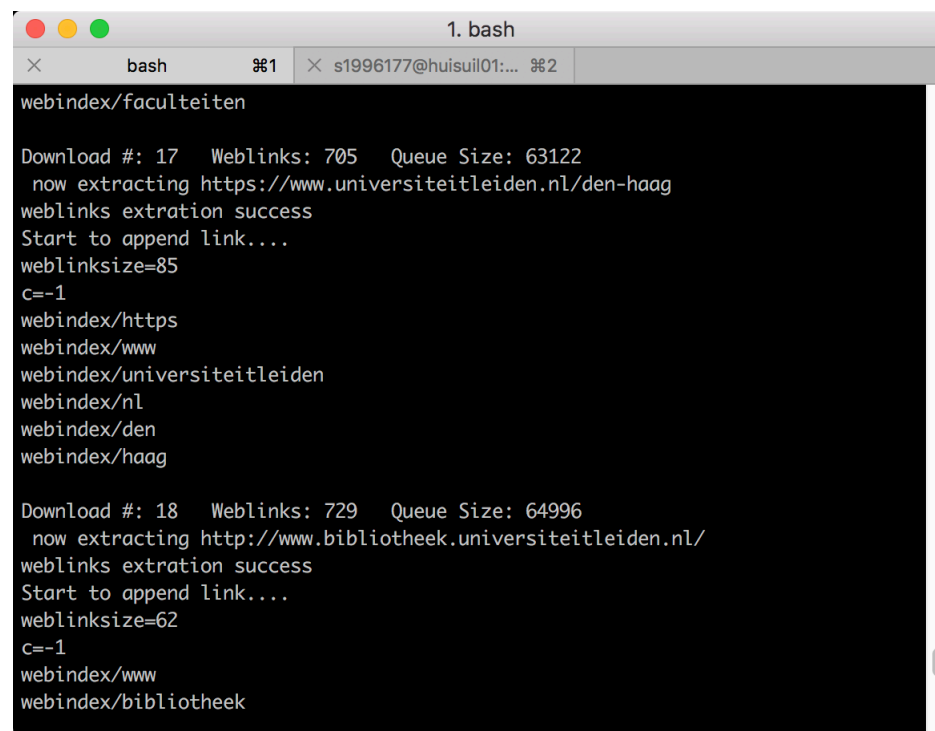
2,a)

Weblinkindex:

specific code please see "void weblinkIndex(char *input)"

My method is to try to get one char by order and record it if it's not separator, but sometimes I would get "http/1.1" in only one time, I wrote a if() function to avoid the problem, but actually I still have no idea why this happens.

Screenshot(in test phase):

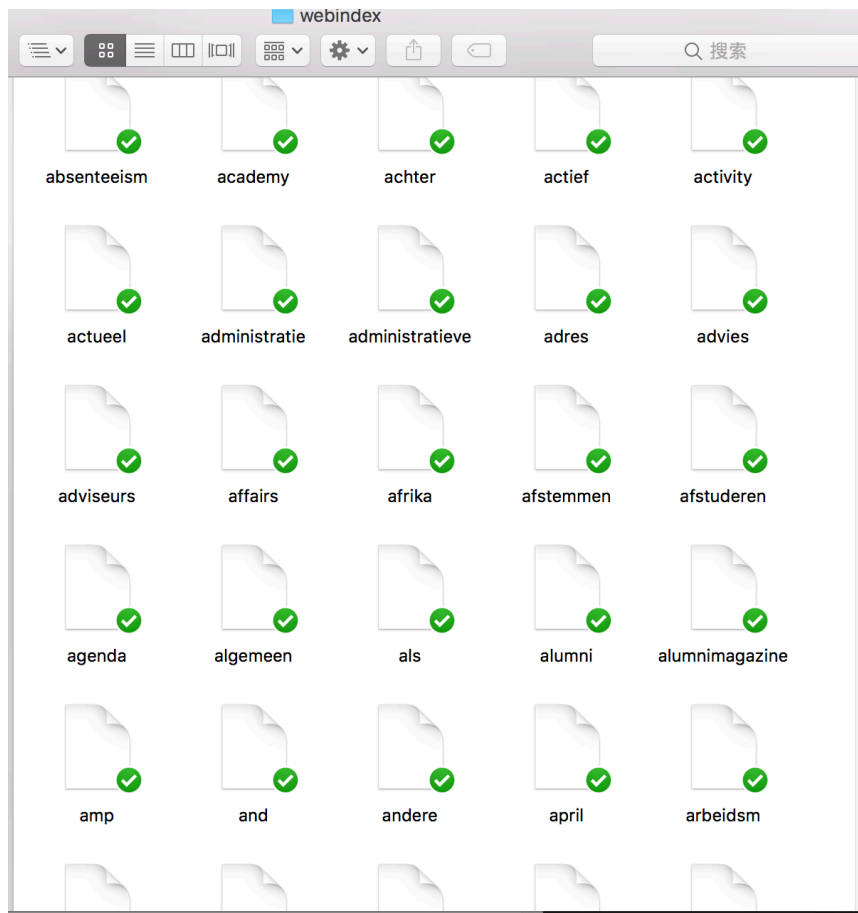


```
webindex/faculteiten

Download #: 17  Weblinks: 705  Queue Size: 63122
now extracting https://www.universiteitleidn.nl/den-haag
weblinks extration success
Start to append link....
weblinksize=85
c=-1
webindex/https
webindex/www
webindex/universiteitleidn
webindex/nl
webindex/den
webindex/haag

Download #: 18  Weblinks: 729  Queue Size: 64996
now extracting http://www.bibliotheek.universiteitleidn.nl/
weblinks extration success
Start to append link....
weblinksize=62
c=-1
webindex/www
webindex/bibliotheek
```

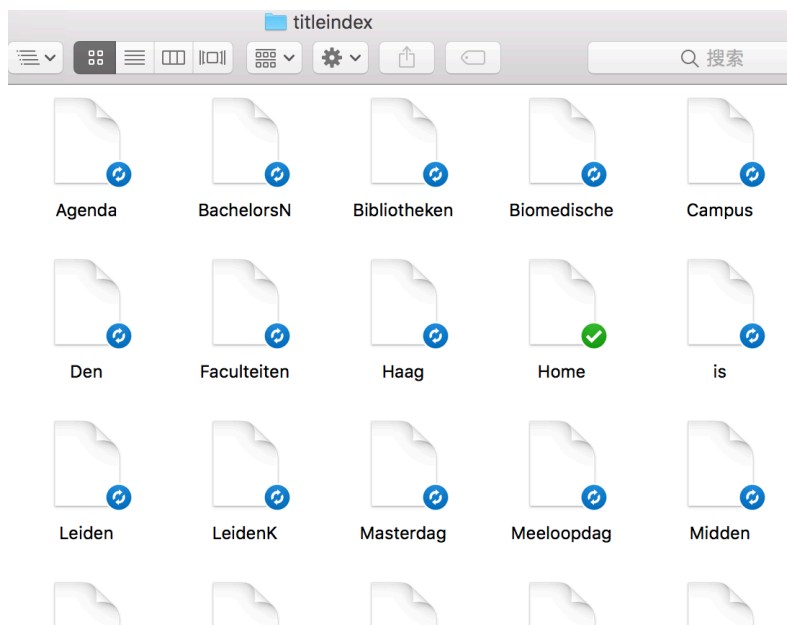
Result:



Title Index:

Specific code please see Function void titleIndex(char *input,char *url,char *type)

Result:



Link index:

Specific code please see Function "void linkIndex(char *url,char *weblinks)"

Here I imported the crc64 algorithm to encrypt the filename of the webpage and the related code are from internet, wrote by a Chinese. The code are download from <http://blog.csdn.net/l1028386804/article/details/50748724>.

Runtime screenshot:

```
indexing weblink=https://www.universiteitleiden.nl/geesteswetenschappen
webindex finished
Linkindex finished

Download #: 41  Weblinks: 1140  Queue Size: 99309
now extracting https://www.universiteitleiden.nl/geneeskunde-lumc
weblinks extration success
Start to append link....
appending links.....we have 71 weblinks to append
indexing weblink=https://www.universiteitleiden.nl/geneeskunde-lumc
webindex finished
Linkindex finished

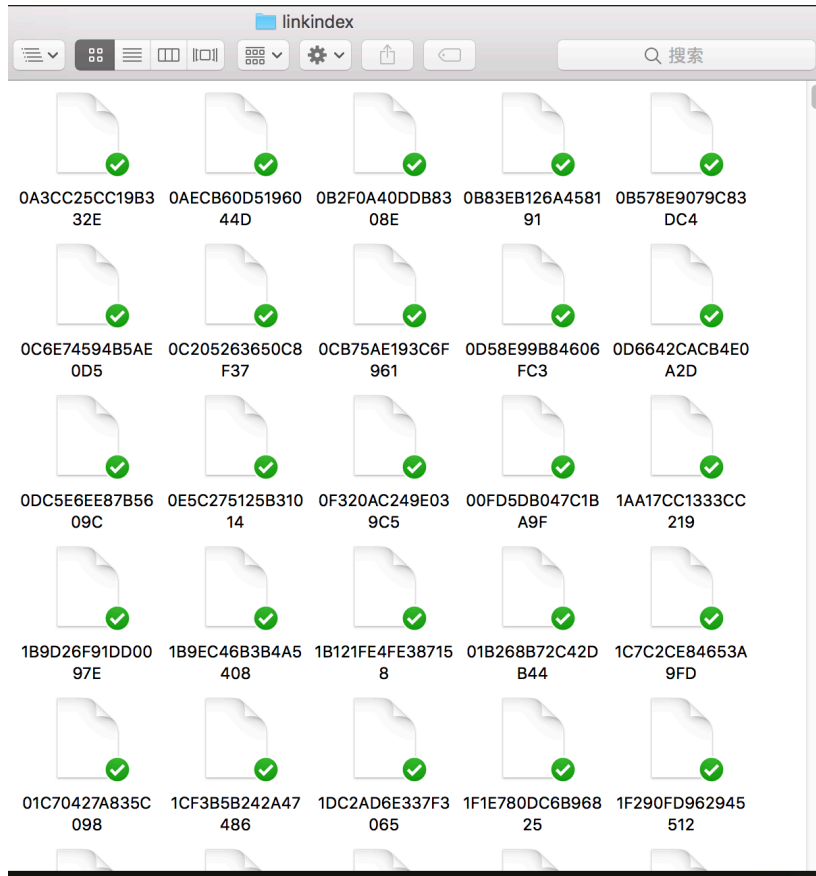
Download #: 42  Weblinks: 1148  Queue Size: 99920
now extracting https://www.universiteitleiden.nl/governance-and-global-affairs
weblinks extration success
Start to append link....
appending links.....we have 103 weblinks to append
indexing weblink=https://www.universiteitleiden.nl/governance-and-global-affairs
webindex finished
Linkindex finished

Download #: 43  Weblinks: 1177  Queue Size: 102137
now extracting https://www.universiteitleiden.nl/rechtsgeleerdheid
weblinks extration success
Start to append link....
appending links.....we have 98 weblinks to append
indexing weblink=https://www.universiteitleiden.nl/rechtsgeleerdheid
webindex finished
Linkindex finished

Download #: 44  Weblinks: 1210  Queue Size: 104746
now extracting https://www.universiteitleiden.nl/sociale-wetenschappen
weblinks extration success
Start to append link....
appending links.....we have 112 weblinks to append
to_add== <li> is not a valid url,will continue...
indexing weblink=https://www.universiteitleiden.nl/sociale-wetenschappen
webindex finished
to_add== <li> is not a valid url,will continue...
Linkindex finished

Download #: 45  Weblinks: 1249  Queue Size: 108074
now extracting https://www.universiteitleiden.nl/wiskunde-en-natuurwetenschappen
```

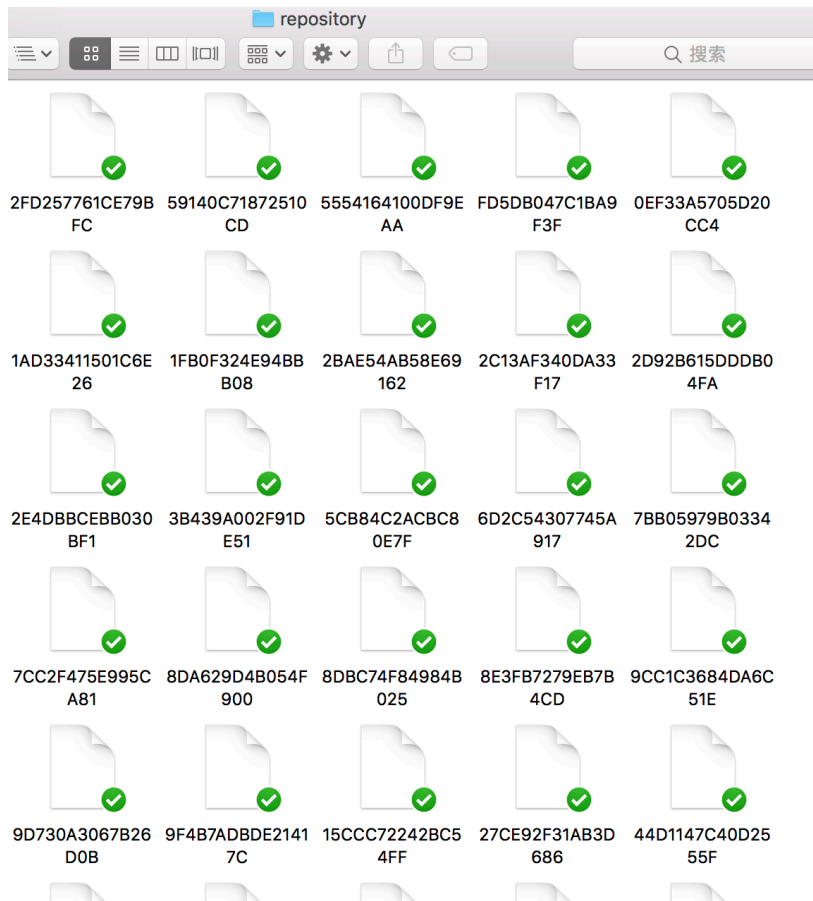
Result:



Repository:

Please see Function "store_webpage(char *buffer,char *myurl)"

All webpage will be stored under "repository/docID" , and the docIDs are also generated by CRC64 algorithm.



Relevance Ranking:

The code of this part has not finished yet, only ideas here. In this part I will build a binary sort tree to achieve the goal of sorting the relevant url/webpage. The structure of the BStree is slightly different with the BStree I used in the Duplicate detection part. Except the relational pointer(left/right child etc.), every tree node will have a char* to store the weblink and a int to store the relevance score.

When we type in a keyword, the program will search the word under titleindex/ first in read in all links in the correspond file, then build tree nodes for each links with a relevance score of 16.

Similar process will be implement in weblinkindex(for 4 relevance score ,etc.)

If we the weblink has been built yet, no new node will be build and only the score will be added on. On the otherside, when the score is modified, it will automatically goto the new position it should be(tree structure rebuild).

And when the process finished ,we can just preorder the whole tree(in case of bigger relevance score is in the leftside.)

2,b)

ImageIndex:

Specific code please see function "void imgindex(char *title,char *imgurl,char *type)"

Every link of image is wrote under those filenames, generated by the title of the webpage.(each word in title would generate a file.)

Result:

