Web and Network Science Assignment 1

Sai Krishna Lakshminarayanan (18230229) ,Surya Balakrishnan Ramakrishnan(18231072)

4 February 2019

1.Introduction:

Web crawling is the process of systematically scanning the world wide web to either extract information from the webpage or to analyse the connectivity of a web page with some other webpage. Web crawling is done through an internet bot which is called spider or crawler. Web crawlers copies the indented web site so that it can be processed by a search engine which indexes the required page which makes it easy for the user to easily search the web page. In this case we have used R programming language with the Rcrawler package which crawls through a given URL to find the connected pages, while on the other hand the Rvest package is used to extract some information from the intended web page.

2.Data crawling methodology:

We have used the web sites of Dublin Business School (dbs.ie) and Letterkenny Institute of Technology (lyit.ie) for the purpose of web crawling. The first step would be to determine the connectivity of both the web sites. The connectivity is determined by crawling the web sites to check its' out degrees in other words we check which are the web pages which are directly linked to. This task is carried out using the Rcrawler package which takes an URL as a reference and crawls through all the pages connected to the reference page. The Rcrawler by default crawls through pages up to level 10 which means that it will keep crawling till it reaches a page which is 10 steps away from the initial reference URL. Due to computation and time constraints we represent the pages until 2 levels, in other words we have crawled pages which are 2 steps away from the reference URL. The crawled web pages are processed and its properties such are in degree, out degree, from node, to node are stored in a data frame. Now for representing the data frame it is necessary that it is converted to the igraph format which is done using graph.data.frame command.

The crawled websites are represented as a directed graph, which is a connection of vertices using edges. Here vertices represent the web pages themselves and edges represent the connectivity between the web pages. Directed graph means that the connectivity is directed and not necessarily omni directional. The R igraph package is used to create a directed graph of the crawled web site. Before plotting the crawled web sites, it is necessary that some data cleaning methodology is implemented so that redundant connections, loop and multiple edges can be removed. This is accomplished using the simplify function of the igraph package.

All of them are saved and stored in the folder as and when they're generated so that they can be used for future processing purposes efficiently. These files are given along with the zip file to be sent.

3.Extracting data from the web pages:

Since we took the web pages of Dublin Business School and Letterkenny Institute of Technology, we decided a comparison could be made between the courses which each of these institutions offers. To crawl through these pages and extract the required information we used the Rvest package. The Rvest along with an add on extension called selector gadget has been used to extract information which are stored within the HTML tags of the respective web pages. To select the course information which is stored within the **TAG NAME** to extract all the names of the courses which are offered by the educational institution. Similarly details such as the course type undergraduate or postgraduate and the NFQ level of the course has also been extracted. Based on the extracted information the course offerings and its type and NFQ levels can be compared.

We have also computed the measure of graph connectivity which represents which of the nodes are actually connected using an edge. Based on the connectivity we have determined which of the nodes are strongly and weekly connected. We also computed the components of the graph. We also computed the shortest distance between the different pages and also the diameter which represents the maximum distance between any two nodes, in this case web pages.

4.Results

4.1Website 1- Dublin Business School

4.1.1Case 1

```
#loading the required packages
library(Rcrawler)

library(igraph)

library(rvest)

# First website Dublin Business School
Rcrawler(Website = "https://www.dbs.ie/", no_cores = 8, no_conn = 8 , NetworkData = TRUE, NetwExtLinks =TRUE
,  statslinks = FALSE,MaxDepth = 1,Timeout = 2)#case 1 with stats link as true

#giving depth as 1 for computational and time constraint reason. provide 2 or 3 if needed but it will take alot of tim
e.

#getting insights of the obtained dataset
head(NetwEdges)

##   From To Weight Type
## 1    1  2      0    1
## 2    1  3      0    1
## 3    1  4      0    1
## 4    1  5      0    1
## 5    1  6      0    1
## 6    1  7      0    1

head(NetwIndex)
```

```
##
##                  "https://www.dbs.ie/"
##                       InternalLinks1
##        "https://www.dbs.ie/cookie-policy"
##                       InternalLinks2
##                   "http://www.dbs.ie/"
##                       InternalLinks3
## "http://www.dbs.ie/about-dbs/for-employers"
##                       InternalLinks4
##  "http://www.dbs.ie/International-students"
##                       InternalLinks5
##       "http://www.dbs.ie/news-and-events"

head(INDEX)

##   Id                              Url    Stats Level OUT IN
## 1  1                  https://www.dbs.ie/ finished     0 106  1
## 2  2        https://www.dbs.ie/cookie-policy finished     1 137  1
## 3  3                   http://www.dbs.ie/ finished     1  83  1
## 4  4 http://www.dbs.ie/about-dbs/for-employers finished     1 112  1
## 5  5  http://www.dbs.ie/International-students finished     1 133  1
## 6  6       http://www.dbs.ie/news-and-events finished     1 126  1
##   Http Resp Content Type Encoding Accuracy
## 1      200    text/html    UTF-8
## 2      200    text/html    UTF-8
## 3      200    text/html    UTF-8
## 4      200    text/html    UTF-8
## 5      200    text/html    UTF-8
## 6      200    text/html    UTF-8

#considering to plot the entire nodes and edges
big<-data.frame(From=NetwEdges$From,To=NetwEdges$To)
big1<-graph.data.frame(big,directed = T)
big1<-simplify(big1, remove.multiple = TRUE, remove.loops = TRUE,
               edge.attr.comb = igraph_opt("edge.attr.comb"))
plot(big1)
```

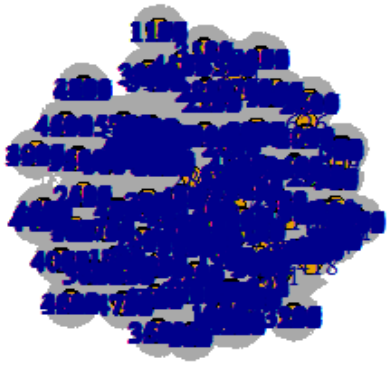

Fig :1 Graph of the uncleaned dataset with all nodes

```
#simplifying and plotting for the in and out degree values for the ones with url scrapped
dbsgraph<-data.frame(IN=INDEX$IN,OUT=INDEX$OUT)
dbsgraph<-graph.data.frame(dbsgraph,directed = T)

dbsgraph<-simplify(dbsgraph, remove.multiple = TRUE, remove.loops = TRUE,
       edge.attr.comb = igraph_opt("edge.attr.comb"))#removing the duplicate and loops

#plotting for the cleaned ones
plot(dbsgraph, edge.arrow.size=0.25, vertex.color="gold", vertex.size=15,

    vertex.frame.color="red", vertex.label.color="black",

    vertex.label.cex=0.7, vertex.label.dist=0.1, edge.curved=0.2)
```

Fig 2: Cleaned Data for case 1 of the DBS website

4.1.2Case 2

In this case, considering that statslink is true for the given dataset.

```
#scenario two for statslink as true
Rcrawler(Website = "https://www.dbs.ie/", no_cores = 8, no_conn = 8 , NetworkData = TRUE, NetwExtLinks =TRUE
,  statslinks = TRUE,MaxDepth = 1,Timeout = 2)

#plot for case 2
dbsnetwork<-data.frame(IN=INDEX$IN,OUT=INDEX$OUT)
dbsnetwork<-graph.data.frame(dbsnetwork,directed = T)

dbsnetwork<-simplify(dbsnetwork, remove.multiple = TRUE, remove.loops = TRUE,
       edge.attr.comb = igraph_opt("edge.attr.comb"))

plot(dbsnetwork, edge.arrow.size=0.25, vertex.color="gold", vertex.size=15,

    vertex.frame.color="red", vertex.label.color="black",

    vertex.label.cex=0.8, vertex.label.dist=0.1, edge.curved=0.2)
```

Fig 3: Cleaned Data for case 2 of the DBS website

4.1.3 Finding Distance, Diameter, Strong and Weak Components for DBS

```
#getting the weak components
weak_dbs <- components(dbsnetwork, mode="weak")
groups(weak_dbs)
```

```
## $`1`
##  [1] "3"   "87"  "85"  "84"  "5"   "1"   "41"  "40"  "39"  "38"  "37"
## [12] "36"  "35"  "34"  "33"  "32"  "31"  "2"   "29"  "11"  "9"   "8"
## [23] "7"   "6"   "4"   "19"  "27"  "24"  "14"  "106" "112" "126" "120"
## [34] "136" "146" "209" "130" "141" "118" "158" "122" "134" "150" "176"
## [45] "115" "113" "123" "114" "131" "55"  "128" "125" "124" "138" "82"
## [56] "121" "119" "108" "111" "116" "148" "72"  "71"
##
## $`2`
## [1] "62"  "137"
##
## $`3`
## [1] "88" "83"
##
## $`4`
## [1] "86"  "15"  "133"
##
## $`5`
## [1] "10"  "127" "139"
##
## $`6`
## [1] "25"  "156"
##
## $`7`
## [1] "23"  "132"
##
```

```
## $`8`
## [1] "13"  "144"
##
## $`9`
## [1] "12"  "207"

#getting strong components
strong_dbs <- components(dbsnetwork, mode="strong")
groups(strong_dbs)

## $`1`
## [1] "12"
##
## $`2`
## [1] "207"
##
## $`3`
## [1] "13"
##
## $`4`
## [1] "144"
##
## $`5`
## [1] "14"
##
## $`6`
## [1] "15"
##
## $`7`
## [1] "23"
##
## $`8`
## [1] "132"
##


farthest_vertices(dbsnetwork,directed = TRUE)

## $vertices
## + 2/81 vertices, named, from c5fb824:
## [1] 3   106
##
## $distance
## [1] 1

#diameter of the graph
diameter(dbsnetwork, directed = TRUE)

## [1] 1
```

4.2 Website 2- Letterkenny Institute of Technology

4.2.1 Case 1

```
#For second case with Letterkenny Institute of Technology with stats link as false in first casse
Rcrawler(Website = "https://www.lyit.ie/", no_cores = 8, no_conn = 8 , NetworkData = TRUE, NetwExtLinks =TRUE,
statslinks = FALSE,MaxDepth = 1,Timeout = 2)

#peeking into the dataset
head(NetwEdges)

##   From To Weight Type
## 1    1  2      0    1
## 2    1  3      0    1
## 3    1  4      0    1
## 4    1  5      0    1
## 5    1  6      0    1
## 6    1  7      0    1

head(NetwIndex)

##
##                "https://www.lyit.ie/"
##                        InternalLinks1
##  "https://www.lyit.ie/PrivacyPolicy.aspx"
##                        InternalLinks2
##              "https://www.lyit.ie/Home"
##                        InternalLinks3
##       "https://www.lyit.ie/Study-at-LYIT"
##                        InternalLinks4
##       "https://www.lyit.ie/Student-Life"
##                        InternalLinks5
## "https://www.lyit.ie/Research-Innovation"

head(INDEX)

##   Id                              Url    Stats Level OUT IN
## 1  1                 https://www.lyit.ie/ finished     0 176  1
## 2  2  https://www.lyit.ie/PrivacyPolicy.aspx finished     1 179  1
## 3  3              https://www.lyit.ie/Home finished     1 177  1
## 4  4      https://www.lyit.ie/Study-at-LYIT finished     1 174  1
## 5  5       https://www.lyit.ie/Student-Life finished     1 172  1
## 6  6 https://www.lyit.ie/Research-Innovation finished     1 171  1
##   Http Resp Content Type Encoding Accuracy
## 1      200    text/html    UTF-8
## 2      200    text/html    UTF-8
## 3      200    text/html    UTF-8
## 4      200    text/html    UTF-8
## 5      200    text/html    UTF-8
## 6      200    text/html    UTF-8

#putting for large one
big<-data.frame(From=NetwEdges$From,To=NetwEdges$To)
big1<-graph.data.frame(big,directed = T)
big1<-simplify(big1, remove.multiple = TRUE, remove.loops = TRUE,
               edge.attr.comb = igraph_opt("edge.attr.comb"))
```

```
plot(big1)
```

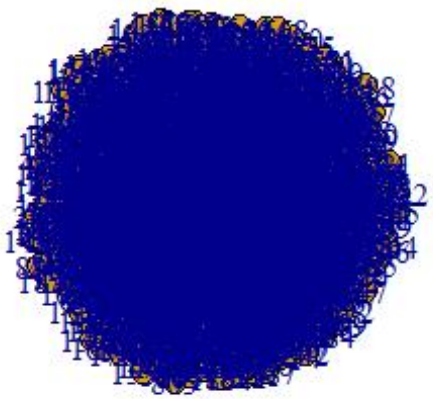

Fig 4: Graph of the uncleaned dataset with all nodes

```
a<-data.frame(IN=INDEX$IN,OUT=INDEX$OUT)
litNetwork <- graph.data.frame(a, directed=T)
lyitINDEX <- INDEX
lyitNetwEdges <- NetwEdges

#cleaning and simplifying the data by removing duplicate and loop
litNetwork<-simplify(litNetwork, remove.multiple = TRUE, remove.loops = TRUE,
       edge.attr.comb = igraph_opt("edge.attr.comb"))

#plotting based on it
plot(litNetwork, edge.arrow.size=0.3, vertex.color="gold", vertex.size=15,

    vertex.frame.color="gray", vertex.label.color="black",

    vertex.label.cex=0.9, vertex.label.dist=0.1, edge.curved=0.5)
```

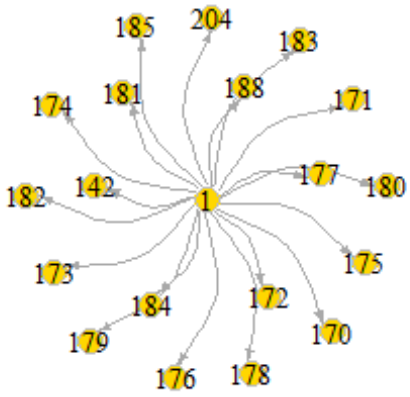

Fig 5: Cleaned Data for case 1 of the LYIT website

4.2.2 Case 2

```
#For second case with Letterkenny Institute of Technology with statslinks as true
Rcrawler(Website = "https://www.lyit.ie/", no_cores = 8, no_conn = 8 , NetworkData = TRUE, NetwExtLinks =TRUE,
statslinks = TRUE,MaxDepth = 1,Timeout = 2)

#plotting it
a<-data.frame(IN=INDEX$IN,OUT=INDEX$OUT)
litNetwork <- graph.data.frame(a, directed=T)
litNetwork<-simplify(litNetwork, remove.multiple = TRUE, remove.loops = TRUE,
       edge.attr.comb = igraph_opt("edge.attr.comb"))
plot(litNetwork, edge.arrow.size=0.3, vertex.color="gold", vertex.size=15,

    vertex.frame.color="gray", vertex.label.color="black",

    vertex.label.cex=0.5, vertex.label.dist=0.1, edge.curved=0.5)
```

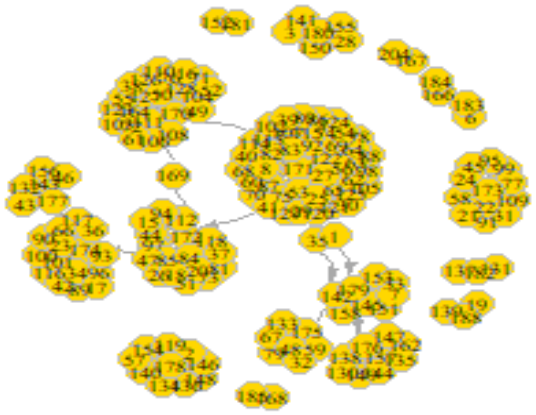

Fig 6: Cleaned Data for case 2 of the LYIT website

4.2.3 Finding Distance, Diameter, Strong and Weak Components for DBS

```
#calcualting farthest vertices
farthest_vertices(litNetwork,directed = TRUE)

## $vertices
## + 2/174 vertices, named, from 8665311:
## [1] 164 177
##
## $distance
## [1] 5

#calculating diameter
diameter(litNetwork, directed = TRUE)

## [1] 5

#getting shortest paths
dist_matrix <- distances(litNetwork,  mode="out")
shortest_paths(litNetwork, from="164", to = "177", mode = c("out"), # out indicates paths from node 12
  weights = NULL,  output = c("both"))

## $vpath
## $vpath[[1]]
## + 6/174 vertices, named, from 8665311:
## [1] 164 170 171 172 174 177
```

```
##
##
## $epath
## $epath[[1]]
## + 5/164 edges from 8665311 (vertex names):
## [1] 164->170 170->171 171->172 172->174 174->177
##
##
## $predecessors
## NULL
##
## $inbound_edges
## NULL
```

```
all_shortest_paths(litNetwork, from="1", to =V(litNetwork), mode = c("out"))
```

```
## $res
## $res[[1]]
## + 4/174 vertices, named, from 8665311:
## [1] 1   171 172 174
##
## $res[[2]]
## + 3/174 vertices, named, from 8665311:
## [1] 1   171 172
##
## $res[[3]]
## + 2/174 vertices, named, from 8665311:
## [1] 1   171
##
## $res[[4]]
## + 1/174 vertex, named, from 8665311:
## [1] 1
##
## $res[[5]]
## + 2/174 vertices, named, from 8665311:
## [1] 1   179
##
## $res[[6]]
## + 5/174 vertices, named, from 8665311:
## [1] 1   171 172 174 177
##
##
## $nrgeo
##   [1] 0 0 1 1 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
##  [36] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
##  [71] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [106] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [141] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0
```

```
#strong components
str <- components(litNetwork, mode="strong")
groups(str)
```

```
## $`1`
## [1] "6"
##
## $`2`
## [1] "183"
##
## $`3`
## [1] "7"
##
## $`4`
```

```
## [1] "8"
##
## $`5`
## [1] "2"
##
## $`6`
## [1] "3"
##
## $`7`
## [1] "15"
##
## $`8`
## [1] "16"
##
## $`9`
## [1] "17"
##
## $`10`
## [1] "18"
##
## $`11`
## [1] "19"
```

```
#weak components
weak <- components(litNetwork, mode="weak")
groups(weak)
```

```
## $`1`
##   [1] "162" "175" "174" "172" "171" "170" "169" "35"  "164" "1"   "158"
##  [12] "157" "156" "153" "151" "149" "147" "145" "144" "143" "142" "138"
##  [23] "135" "133" "132" "130" "129" "128" "127" "126" "125" "124" "123"
##  [34] "122" "121" "120" "118" "117" "116" "115" "114" "113" "112" "111"
##  [45] "110" "108" "107" "106" "105" "104" "103" "102" "101" "100" "98"
##  [56] "97"  "96"  "94"  "93"  "92"  "90"  "89"  "88"  "87"  "86"  "85"
##  [67] "84"  "83"  "82"  "81"  "80"  "79"  "78"  "76"  "75"  "74"  "73"
##  [78] "72"  "71"  "70"  "69"  "68"  "67"  "66"  "65"  "64"  "63"  "62"
##  [89] "61"  "60"  "59"  "56"  "55"  "54"  "53"  "52"  "51"  "50"  "49"
## [100] "48"  "47"  "46"  "44"  "43"  "42"  "41"  "40"  "39"  "38"  "37"
## [111] "36"  "34"  "33"  "32"  "30"  "29"  "27"  "26"  "25"  "23"  "20"
## [122] "18"  "17"  "16"  "15"  "8"   "7"   "176" "179" "177"
##
## $`2`
## [1] "168" "185"
##
## $`3`
## [1] "167" "204"
##
## $`4`
## [1] "166" "184"
##
## $`5`
## [1] "155" "150" "141" "28"  "3"   "180"
##
## $`6`
##  [1] "154" "148" "146" "140" "136" "134" "119" "57"  "2"   "178"
##
## $`7`
## [1] "152" "181"
##
## $`8`
## [1] "139" "19"  "188"
##
## $`9`
## [1] "137" "131" "182"
##
## $`10`
##  [1] "109" "99"  "95"  "91"  "77"  "58"  "45"  "31"  "24"  "22"  "21"
## [12] "173"
##
## $`11`
## [1] "6"   "183"
```

4.3 Scrapping the course data from the DBS and LYIT websites:

```
#Scrapping the data regarding dbs ug courses
url_dbs<-read_html('https://www.dbs.ie/courses/full-time-undergraduate')
dbs_course<-data.frame(Name= html_text(html_nodes(url_dbs,".CourseListItem h4 a")),
                       Type= html_text(html_nodes(url_dbs,".CourseType span")),
                       Level=html_text(html_nodes(url_dbs,".CourseType+ div span"))
                       )

url_lyit<-read_html('https://www.lyit.ie/Study-at-LYIT/Find-a-course/?lvl=4')
lyit_course<-data.frame(Name=          html_text(html_nodes(url_lyit,".CourseListItem          h3")),
                        Type=          html_text(html_nodes(url_lyit,".CourseType              span")),
                        Level=html_text(html_nodes(url_lyit,".CourseType+          div          p"))
                        )
```

	Name	Type	Level
1	BA (Hons) Accounting & Finance – Full-time	International, Full-time Degrees/ Certificates	Full-Time
2	BA (Hons) Audio Production & Music Project Management	International, Full-time Degrees/ Certificates	Full-Time
3	BA (Hons) Business – Full-time	International, Full-time Degrees/ Certificates	Full-Time
4	BA (Hons) Business (HRM) – Full-time	International, Full-time Degrees/ Certificates	Full-Time
5	BA (Hons) Business (Law) – Full-time	International, Full-time Degrees/ Certificates	Full-Time
6	BA (Hons) Business (Management) – Full-time	International, Full-time Degrees/ Certificates	Full-Time
7	BA (Hons) Business (Project Management) – Full-time	International, Full-time Degrees/ Certificates	Full-Time
8	BA (Hons) Business (Psychology) - Full-Time	International, Full-time Degrees/ Certificates	Full-Time
9	BA (Hons) Business (Work Placement) – Full-time	International, Full-time Degrees/ Certificates	Full-Time
10	BA (Hons) Business Information Systems – Full-time	International, Full-time Degrees/ Certificates	Full-Time
11	BA (Hons) Business Information Systems (Cloud Computi...	International, Full-time Degrees/ Certificates	Full-Time
12	BA (Hons) Film – Full-time	International, Full-time Degrees/ Certificates	Full-Time
13	BA (Hons) Financial Services – Full-time	International, Full-time Degrees/ Certificates	Full-Time
14	BA (Hons) in Applied Social Care	Full-time Degrees/ Certificates	Full-Time
15	BA (Hons) Marketing – Full-time	International, Full-time Degrees/ Certificates	Full-Time

Fig 7: Scrapped data of Courses from DBS website

5.Observations:

- The website of Letterkenny Institute of Technology has more weak nodes from which we can say that this website is weakly connected, in other words there does not exist a path between every pair of nodes.
- The website of Dublin Business School has a greater number of strong nodes which means that it is strongly connected and there exist a path between every pair of nodes.
- The number of out degrees for the Dublin Business School's website which means there are more pages which are linked to it as compared to the website of Dublin Business School.
- Even though Letterkenny Institute of Technology's website is properly connected and provides more information, the construction of the website was not proper, the use of inline HTML editing causes difficulty to crawl the web site and obtain information from it, whereas the web site of Dublin Business School is properly tagged and structured.

- The Dublin Business School's website has proper name given for each of the information such as .CourseListItem h4 a which holds the name of the course, whereas the Letterkenny Institute of Technology's website gives generic reference to the information such as .CouseInfo h4.
- From the graph it is evident that the interconnectivity is better in the Dublin Business School's website.

6.Conclusion:

From the graphical representations and several mathematical metrics computed we can say that the website of Dublin Business School is well connected and has more pages linked to it which suggests that the amount of information which one can find on its webpage is more as compared to the website of Letterkenny Institute of Technology.

7.Work Split Up:

7.1Work done by Sai Krishna Lakshminarayanan (18230229)

- Crawling the website of Dublin Business School (DBS).
- Extracting information from the website of DBS.
- Construction of graphs using igraph for DBS site.
- Interpretation of network data for DBS.
- Observations and calculations for DBS site.
- Scrapping of the course data for DBS website.
- Introduction, Methodology and Extraction written part in the document

7.2Work done by Surya Balakrishnan Ramakrishnan (18231072)

- Crawling the web site of Letterkenny Institute of Technology (LYIT).
- Extracting information from the website of LYIT.
- Construction of graphs using igraph for LYIT site.
- Interpretation of network data for LYIT.
- Observations and calculations for LYIT page.
- Scrapping of the course data for LYIT website.
- Observation and Conclusion written part in the document.

8.References:

1. https://www.rdocumentation.org/packages/Rcrawler/versions/0.1.5
2. https://igraph.org/r/doc/simplify.html
3. http://cneurocvs.rmki.kfki.hu/igraph/doc/R/graph.data.frame.html
4. https://cran.r-project.org/web/packages/Rcrawler/Rcrawler.pdf
5. https://en.wikipedia.org/wiki/Distance_(graph_theory)
6. https://en.wikipedia.org/wiki/Web_crawler
7. Week 2 and 3 Lecture Notes in Blackboard
8. https://stackoverflow.com/questions/11784980/how-do-you-build-a-graph-from-a-data-frame-using-the-igraph-package
9. https://www.youtube.com/watch?v=82s8KdZt5v8&t=595s
10. https://www.analyticsvidhya.com/blog/2017/03/beginners-guide-on-web-scraping-in-r-using-rvest-with-hands-on-knowledge/