1. Briefly explain your code architecture and lessons learned. Using Part 3, show a complete trace with 1M input URLs.

There are two important files the .cpp file which contains the main function and is the starting point of the code . Here I parse the url to extract the host port fragments request and path. This is then passed to a Socket Instance defined in Socket.h for Connecting through sockets . For Connection we start with filling getaddrinfo and then use this information to initialize our socket. Then a fresh connection is made for HEAD request to obtain information about robots.txt and then the if the HEAD request is 4xx we make A GET Request

Trace:

```
 360]579Q     0    237E     0H   570D  420I-858993413R    1C   21L
**Crawling 162 ..@2 Mbps
 362]579Q     0    237E     0H   570D  420I-858993413R    1C   21L
**Crawling 162 ..@2 Mbps
 364]579Q     0    237E     0H   570D  420I-858993413R    1C   21L
**Crawling 162 ..@2 Mbps
 366]579Q     0    237E     0H   570D  420I-858993413R    1C   21L
**Crawling 162 ..@2 Mbps
 368]579Q     0    237E     0H   570D  420I-858993413R    1C   21L
**Crawling 162 ..@2 Mbps
 370]579Q     0    237E     0H   570D  420I-858993413R    1C   21L
**Crawling 162 ..@2 Mbps
 372]579Q     0    237E     0H   570D  420I-858993413R    1C   21L
**Crawling 162 ..@2 Mbps
 374]579Q     0    237E     0H   570D  420I-858993413R    1C   21L
**Crawling 162 ..@2 Mbps
 376]579Q     0    237E     0H   570D  420I-858993413R    1C   21L
**Crawling 162 ..@2 Mbps
 378]579Q     0    237E     0H   570D  420I-858993413R    1C   21L
**Crawling 162 ..@2 Mbps
 380]579Q     0    237E     0H   570D  420I-858993413R    1C   21L
**Crawling 162 ..@2 Mbps
 382]579Q     0    237E     0H   570D  420I-858993413R    1C   21L
**Crawling 162 ..@2 Mbps
 384]579Q     0    237E     0H   570D  420I-858993413R    1C   21L
**Crawling 162 ..@1 Mbps
 386]579Q     0    237E     0H   570D  420I-858993413R    1C   21L
**Crawling 162 ..@1 Mbps
 388]579Q     0    237E     0H   570D  420I-858993413R    1C   21L
**Crawling 162 ..@1 Mbps
 390]579Q     0    237E     0H   570D  420I-858993413R    1C   21L
**Crawling 162 ..@1 Mbps
```

5. How many of the crawled 2xx pages contain a hyperlink to our domain tamu.edu? How many of them originate from outside of TAMU? Explain how you obtained this information. Examples of suitable links:

Links found 40

This can be directly found from the HTML Parser.If the page loads correctly we can parse the links to find hosts ending with tamu.edu. If the host of the starting connection is also in tamu.edu it is an internal link otherwise external link

4. What is the probability that a link in the input file contains a unique host? What is the probability that a unique host has a valid DNS record? What percentage of contacted sites had a 4xx robots file?

The probability that the link contains a unique host is very low. The url serves as the anchor text for all its links. The webgraph of the url consists of these links so it is highly unlikely that it contains a unique host.

3.) Obtain the average number of links per HTML page that came back with a 2xx code. Estimate the size of Google's webgraph (in terms of edges and bytes) that contains 1T crawled nodes and all of their out-links. Assume the graph is stored using adjacency lists, where each URL is represented by a 64-bit hash

Average number of links /html page = total links accessed by shared Parameter of threads/number of http pages with code 200 again accessed by Parameters

When we access google.com we get all the links that can be accessed from the page. And this will represent the google webgraph

2.(5 pts) Determine the average page size in bytes (across all HTTP codes). Estimate the bandwidth (in Gbps) needed for Yahoo to crawl 10B pages a day.

Average page size can be found out by dividing the total number of bytes divided by the number of links. This when found out multiplied by 10 B pages will give bandwidth