# CSP 301 Report of Group 5 for PART B

– Abhinav Deep Singh, 2012CS10207
–Chirag Agrawal, 2012CS50285
–Shivam Garg, 2012CS10252

***Corner Cases Of Perl***

- **For Query 1:** “Clique” must be present.
- **For Query 2:** “Shortest” must be present and, “Path” and “length”, or “distance” must be present and two ids and two universities are provided. If more than two ids or universities are given then only the first two are considered.
- **For Query 3:** “Shortest” and “Path” must be present and “list ”, “people”, “person ” are optional and two ids and two universities provided.
- **For Query 4:** “Shortest” must be present and “Path ” is optional and no ids and universities are provided.
- **For Query 5:** “Importance” or “Important ” must be present,( or “Shortest” must be present and “Path ” is optional) and exactly one id and one university are provided.
- **For Query 6:**“Importance” or “Important ” must be present,( or “Shortest” must be present and “Path ” is optional,) and exactly one id and one university are provided, both alongwith “friends or friend”.

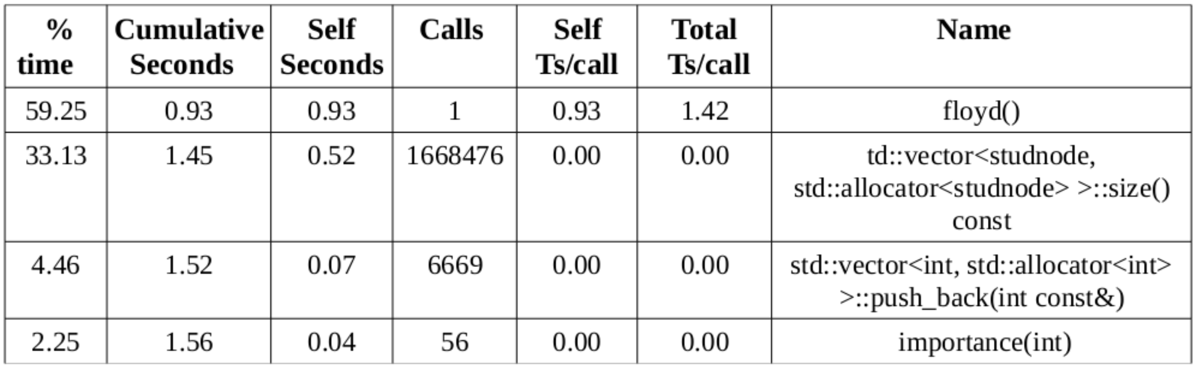
***We ran Valgrind utility on our code and following output came. The analysis of the report is given thereafter.***

```
**-------------------------------------------VALGRIND OUTPUT-------------------------------------------
**------------------------------------------------BEGIN------------------------------------------------

cs1120207@varali:~/CSP_Product2$ valgrind --leak-check=full ./run2.sh
==21626== Memcheck, a memory error detector
==21626== Copyright (C) 2002-2011, and GNU GPL'd, by Julian Seward et al.
==21626== Using Valgrind-3.7.0 and LibVEX; rerun with -h for copyright info
==21626== Command: ./run2.sh
==21626==
Available Universities :: IITD IITK IITB
Processing Data
after exec
Enter Your Query :: clique iitd 123
#----------------------------------------------------------------------------------------INTERPRETED QU
 QUERY ::  1. Size of Clique of a given person (identified by university name and ID)

PERSON 1    =>
 UNIVERSITY  :: IITD
 ID          :: 123

PERSON 2    =>
 UNIVERSITY  :: NOT REQUIRED
 ID          :: NOT REQUIRED
#------------------------------------------------------------------------------------------------x------
```

```
DO YOU WANT TO ANALYZE THIS INTERPRETED QUERY (Y/N) ?
y
#------------------------------------------------------------------------------------------------x-----
Clique size is 4
Members of clique are :
Kshitij Patel
Ravelojaona Jeritiana
Poojan Shree
Ajay Kamal
#------------------------------------------------------------------------------------------------x-----
Enter Your Query :: shortest distance 123 iitd 124 iitb
#----------------------------------------------------------------------------------------INTERPRETED QU
 QUERY ::  2. The length of the shortest path between two given people

PERSON 1    =>
 UNIVERSITY  :: IITD
 ID          :: 123

PERSON 2    =>
 UNIVERSITY  :: IITB
 ID          :: 124
#------------------------------------------------------------------------------------------------x-----
DO YOU WANT TO ANALYZE THIS INTERPRETED QUERY (Y/N) ?
y
#------------------------------------------------------------------------------------------------x-----
Shortest distance between Kshitij Patel and Yash Saxena is 2
#------------------------------------------------------------------------------------------------x-----
Enter Your Query :: shortest path iitd 123 iitb 124
#----------------------------------------------------------------------------------------INTERPRETED QU
 QUERY ::  3. The list of people on the shortest path between two given people

PERSON 1    =>
 UNIVERSITY  :: IITD
 ID          :: 123

PERSON 2    =>
 UNIVERSITY  :: IITB
 ID          :: 124
#------------------------------------------------------------------------------------------------x-----
DO YOU WANT TO ANALYZE THIS INTERPRETED QUERY (Y/N) ?
y
#------------------------------------------------------------------------------------------------x-----
both Kshitij and Tata are friends
27 both Tata and Yash are friends
#------------------------------------------------------------------------------------------------x-----
Enter Your Query :: shortest path
#----------------------------------------------------------------------------------------INTERPRETED QU
 QUERY ::  4. The shortest path in the graph (between any pair of people)

PERSON 1    =>
 UNIVERSITY  :: NOT REQUIRED
 ID          :: NOT REQUIRED
```

```
PERSON 2    =>
 UNIVERSITY  :: NOT REQUIRED
 ID          :: NOT REQUIRED
#------------------------------------------------------------------------------------------------x-----
DO YOU WANT TO ANALYZE THIS INTERPRETED QUERY (Y/N) ?
y
#------------------------------------------------------------------------------------------------x-----
The maximum of shortest path is : 4
#------------------------------------------------------------------------------------------------x-----
Enter Your Query :: importance iitd 123
#----------------------------------------------------------------------------------------INTERPRETED QU
 QUERY ::  5. The importance of the given person (the importance is the number of the all-pair shortest

PERSON 1    =>
 UNIVERSITY  :: IITD
 ID          :: 123

PERSON 2    =>
 UNIVERSITY  :: NOT REQUIRED
 ID          :: NOT REQUIRED
#------------------------------------------------------------------------------------------------x-----
DO YOU WANT TO ANALYZE THIS INTERPRETED QUERY (Y/N) ?
y
#------------------------------------------------------------------------------------------------x-----
Importance of Kshitij Patel is 1029
#------------------------------------------------------------------------------------------------x-----
Enter Your Query :: importance friends 123 iitd
#----------------------------------------------------------------------------------------INTERPRETED QU
 QUERY ::  6. Is any of the friends of a given person more important that him/her

PERSON 1    =>
 UNIVERSITY  :: IITD
 ID          :: 123

PERSON 2    =>
 UNIVERSITY  :: NOT REQUIRED
 ID          :: NOT REQUIRED
#------------------------------------------------------------------------------------------------x-----
DO YOU WANT TO ANALYZE THIS INTERPRETED QUERY (Y/N) ?
y
#------------------------------------------------------------------------------------------------x-----
Importance of Kshitij Patel is 1029
Importance of Kunal Mehrish is 989
Importance of Ravelojaona Jeritiana is 2138
Importance of Divyansh Budumuru is 1731
Importance of Nishant Radhakrishna is 1543
Importance of Prateek Sandhu is 1177
Importance of Akshay Nagpal is 2079
Importance of Daniel Goel is 1946
Importance of Shagun Beniwal is 1312
Importance of Rayala Holm is 1419
Importance of Ishaan Baviskar is 1757
Importance of Raghuvansh Zala is 359
Importance of Poojan Shree is 1424
Importance of Suman Marda is 1516
Importance of Ajay Kamal is 1917
Importance of Tata Agrawal is 1836
Importance of Dhruvin Singh is 1477
Importance of Kishore Agrawal is 1905
Importance of Poojan Batta is 1874
Importance of Kaushal Brahmchari is 1606
Importance of Bjorn Paroda is 1857
Importance of Akash Tripathi is 2179
Importance of Aditya Kumar is 1023
Importance of Namita Bawa is 2359
Importance of Tushar Sandhu is 1552
Importance of Prakash Kailas is 1381
Importance of Rajat Deep is 1618
friends having more importance are :
Ravelojaona Jeritiana entry no.: 112
Divyansh Budumuru entry no.: 147
Nishant Radhakrishna entry no.: 143
Prateek Sandhu entry no.: 131
Akshay Nagpal entry no.: 64
Daniel Goel entry no.: 3
Shagun Beniwal entry no.: 43
Rayala Holm entry no.: 35
Ishaan Baviskar entry no.: 80
Poojan Shree entry no.: 87
Suman Marda entry no.: 121
Ajay Kamal entry no.: 74
Tata Agrawal entry no.: 28
Dhruvin Singh entry no.: 90
Kishore Agrawal entry no.: 13
Poojan Batta entry no.: 46
Kaushal Brahmchari entry no.: 58
Bjorn Paroda entry no.: 92
Akash Tripathi entry no.: 95
Namita Bawa entry no.: 114
Tushar Sandhu entry no.: 65
Prakash Kailas entry no.: 100
Rajat Deep entry no.: 180
#------------------------------------------------------------------------------------------------x-----
Enter Your Query :: quit
==21626==
==21626== HEAP SUMMARY:
==21626==     in use at exit: 1,734 bytes in 50 blocks
==21626==   total heap usage: 51 allocs, 1 frees, 1,750 bytes allocated
==21626==
==21626== LEAK SUMMARY:
==21626==    definitely lost: 0 bytes in 0 blocks
==21626==    indirectly lost: 0 bytes in 0 blocks
==21626==      possibly lost: 0 bytes in 0 blocks
```

```
==21626==    still reachable: 1,734 bytes in 50 blocks
==21626==         suppressed: 0 bytes in 0 blocks
==21626== Reachable blocks (those to which a pointer was found) are not shown.
==21626== To see them, rerun with: --leak-check=full --show-reachable=yes
==21626==
==21626== For counts of detected and suppressed errors, rerun with: -v
==21626== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 2 from 2)



**-------------------------------------------------END------------------------------------------------
```



| % time | Cumulative Seconds | Self Seconds | Calls | Self Ts/call | Total Ts/call | Name |
|---|---|---|---|---|---|---|
| 59.25 | 0.93 | 0.93 | 1 | 0.93 | 1.42 | floyd() |
| 33.13 | 1.45 | 0.52 | 1668476 | 0.00 | 0.00 | td::vector<studnode, std::allocator<studnode> >::size() const |
| 4.46 | 1.52 | 0.07 | 6669 | 0.00 | 0.00 | std::vector<int, std::allocator<int> >::push_back(int const&) |
| 2.25 | 1.56 | 0.04 | 56 | 0.00 | 0.00 | importance(int) |

[?]

Figure 1: Profiler Part A Output

```
**-------------------------------------------VALGRIND REPORT OF PART B & ANALYSIS----------------------
**------------------------------------------------BEGIN------------------------------------------------
The report shows that there was no leaks of "definitely lost", "indirectly lost" or "possibly lost" type

The heap summary result does not change.
There are no errors in the valgrind report.


**-------------------------------------------------END------------------------------------------------
//Same code was tested with profiler.
**-------------------------------------------PROFILER REPORT OF PART B & ANALYSIS----------------------
**------------------------------------------------BEGIN------------------------------------------------
1. The floyd() takes up major chunk of time.(Order n^3)
2. Making the graph takes up next largest time.(Order n^2)
3. The importance() takes up most of the rest time.
4. Rest all  functions takes almost no time.

**-------------------------------------------------END------------------------------------------------
```