Assignment 04 : Analysis of Computer Algorithms’

Ganesh Gundekarla
11700551

1. Establishing the connection between filezilla, cell machine

Process on how implementation is done for MacOS:

Required apps:

a. FileZilla.
b. Cisco any connect client. (vpn).
c. Terminal to access cell machine.

Step one: Using vpn to connect to the unt server:
Here, we had used vpn to connect to the vpn.unt.edu server, to access the cell machine which is the primary step in order to access the cell machine in ubuntu.

Step two: Connection to a cell machine in the unt server .
In terminal we gotta type :

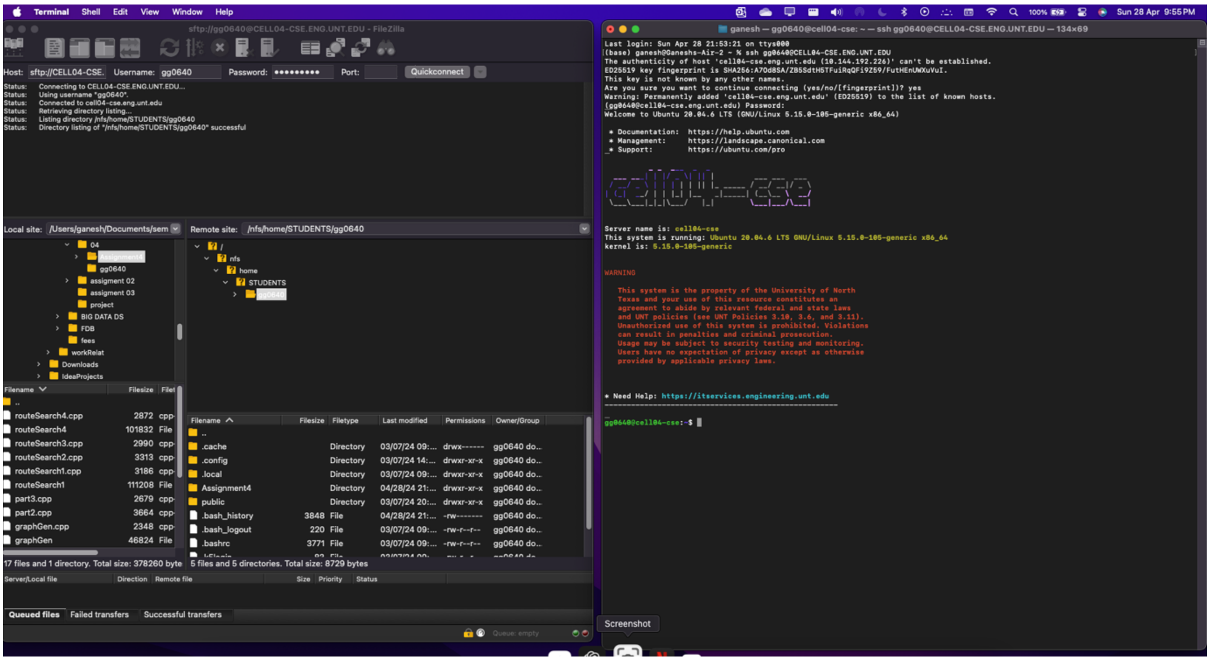
```
ssh gg0640@ CELL04-CSE.ENG.UNT.EDU
```

This will allow us to access a cell machine in the unt server by using our Euid login credentials. After some time, the server gets initiated.

Step 03: Connection of a ftp file transfer protocol with the server .
This step is important in order to upload the algorithm files we had prepared to the cell machine to set up and try out with multiple inputs.

a. Here, in the File zilla app we need to set the HOST =” CELL06-CSE.ENG.UNT.EDU”
b. Username: gg0640
c. Password: my credentials

With this the connection to the server will be complete. With this we will be able to set up a file sharing connection with the server.
The below screenshot shows the successful connection.



sftp://gg0640@CELL04-CSE.ENG.UNT.EDU - FileZilla

Host: sftp://CELL04-CSE. Username: gg0640 Password: ●●●●●●●●● Port: Quickconnect

```
Status:     Connecting to CELL04-CSE.ENG.UNT.EDU...
Status:     Using username "gg0640".
Status:     Connected to cell04-cse.eng.unt.edu
Status:     Retrieving directory listing...
Status:     Listing directory /nfs/home/STUDENTS/gg0640
Status:     Directory listing of "/nfs/home/STUDENTS/gg0640" successful
```

Local site: /Users/ganesh/Documents/sem

Remote site: /nfs/home/STUDENTS/gg0640

Filename	Filesize	Filet
..		
routeSearch4.cpp	2872	cpp-
routeSearch4	101832	File
routeSearch3.cpp	2990	cpp-
routeSearch2.cpp	3313	cpp-
routeSearch1.cpp	3186	cpp-
routeSearch1	111208	File
part3.cpp	2679	cpp-
part2.cpp	3664	cpp-
graphGen.cpp	2348	cpp-
graphGen	46824	File

17 files and 1 directory. Total size: 378260 byte

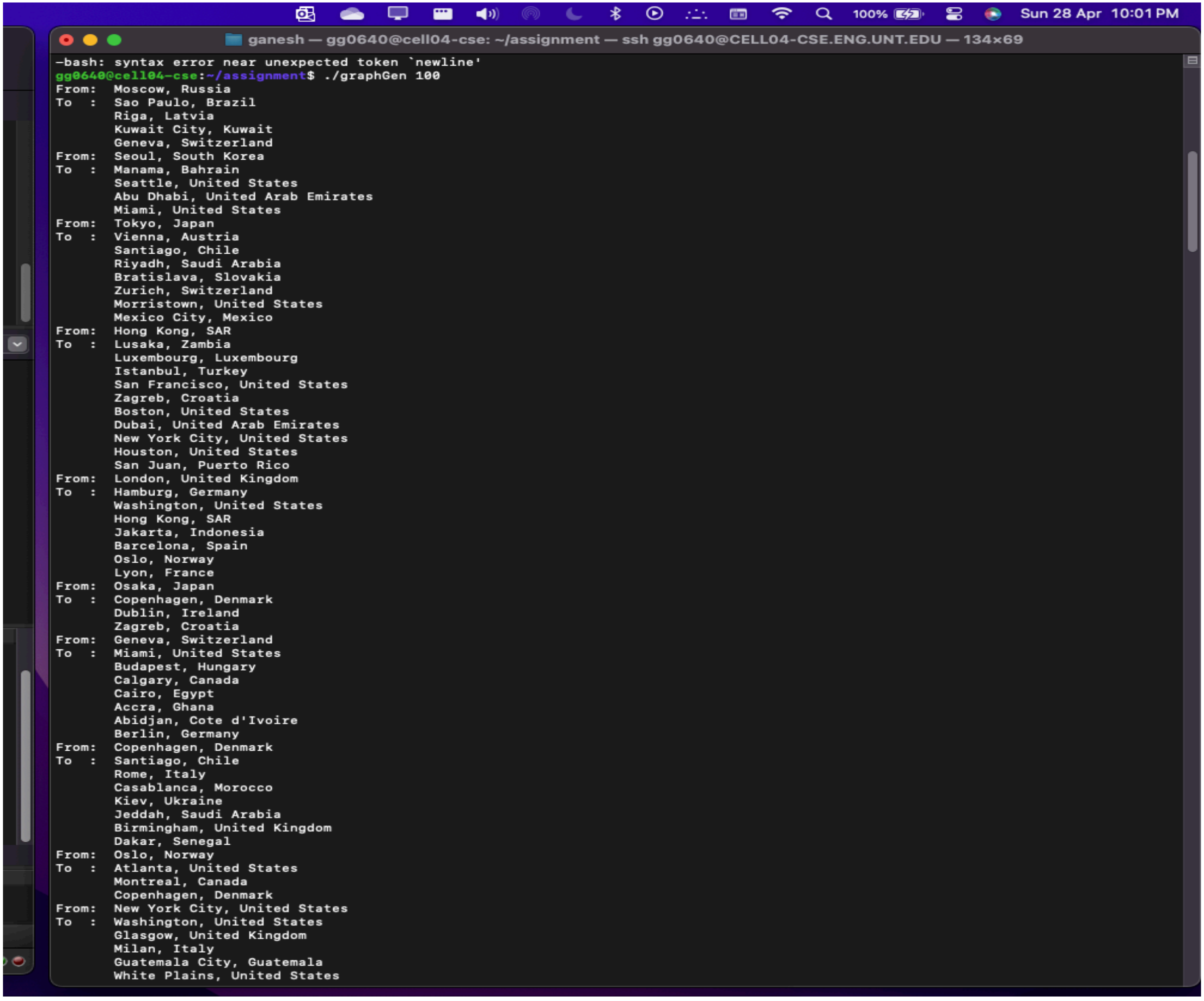
Filename	Filesize	Filetype	Last modified	Permissions	Owner/Group
..					
.cache		Directory	03/07/24 09:...	drwx------	gg0640 do...
.config		Directory	03/07/24 14:...	drwxr-xr-x	gg0640 do...
.local		Directory	03/07/24 09:...	drwxr-xr-x	gg0640 do...
Assignment4		Directory	04/28/24 21:...	drwxr-xr-x	gg0640 do...
public		Directory	03/07/24 20:...	drwxr-xr-x	gg0640 do...
.bash_history	3848	File	04/28/24 21:...	-rw-------	gg0640 do...
.bash_logout	220	File	03/07/24 09:...	-rw-r--r--	gg0640 do...
.bashrc	3771	File	03/07/24 09:...	-rw-r--r--	gg0640 do...

5 files and 5 directories. Total size: 8729 bytes

ganesh — gg0640@cell04-cse: ~ — ssh gg0640@CELL04-CSE.ENG.UNT.EDU — 134×69

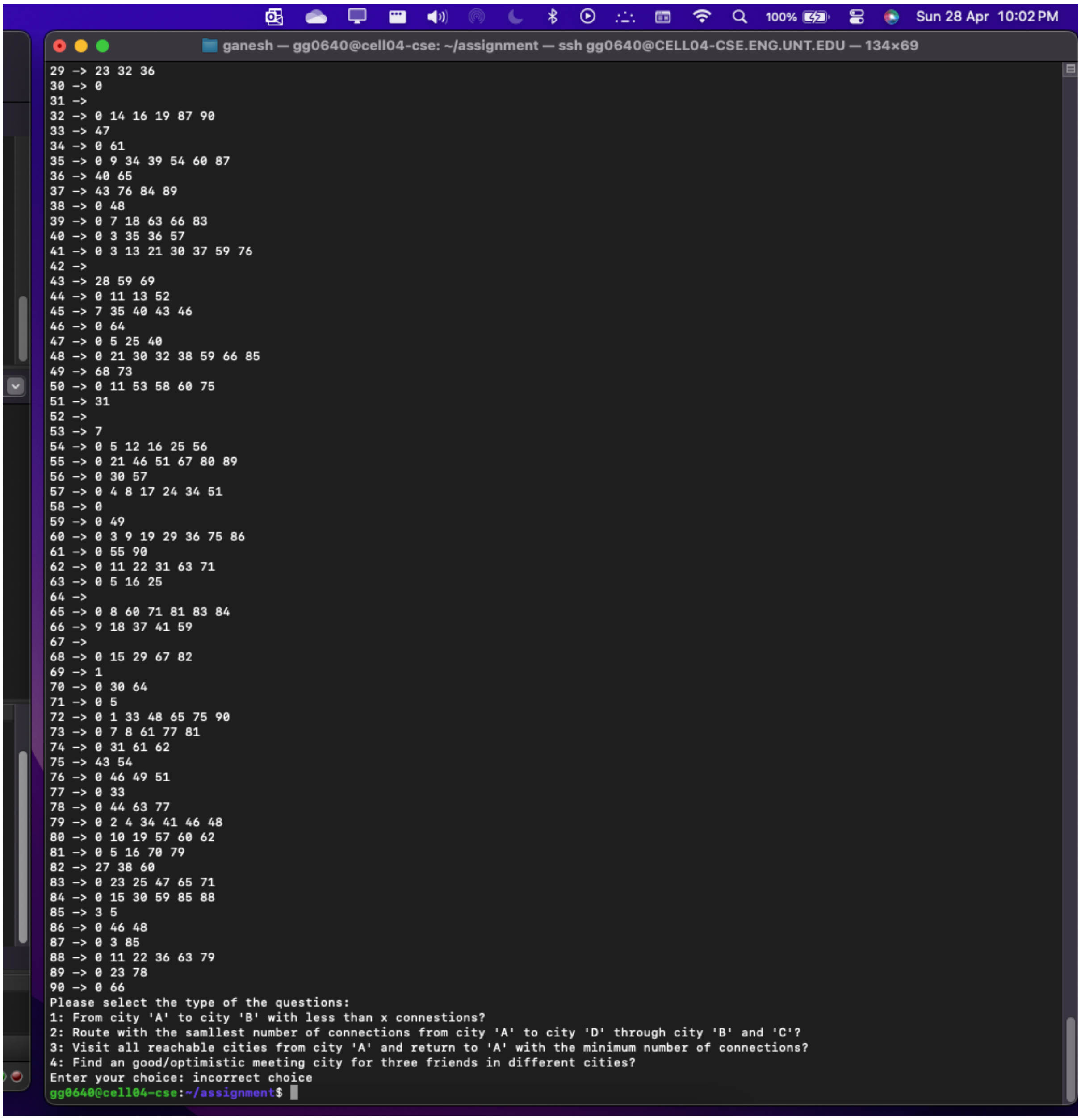
```
Last login: Sun Apr 28 21:53:21 on ttys000
(base) ganesh@Ganeshs-Air-2 ~ % ssh gg0640@CELL04-CSE.ENG.UNT.EDU
The authenticity of host 'cell04-cse.eng.unt.edu (10.144.192.226)' can't be established.
ED25519 key fingerprint is SHA256:A7Od8SA/ZB5SdtH5TFuiRqQFi9Z59/FutHEnUWXuVuI.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'cell04-cse.eng.unt.edu' (ED25519) to the list of known hosts.
(gg0640@cell04-cse.eng.unt.edu) Password:
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.15.0-105-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

Server name is: cell04-cse
This system is running: Ubuntu 20.04.6 LTS GNU/Linux 5.15.0-105-generic x86_64
kernel is: 5.15.0-105-generic

WARNING

   This system is the property of the University of North
   Texas and your use of this resource constitutes an
   agreement to abide by relevant federal and state laws
   and UNT policies (see UNT Policies 3.10, 3.6, and 3.11).
   Unauthorized use of this system is prohibited. Violations
   can result in penalties and criminal prosecution.
   Usage may be subject to security testing and monitoring.
   Users have no expectation of privacy except as otherwise
   provided by applicable privacy laws.

* Need Help: https://itservices.engineering.unt.edu
-----------------------------------------------------

gg0640@cell04-cse:~$ 
```

2. Listing all the cities from the graphgen.



ganesh — gg0640@cell04-cse: ~/assignment — ssh gg0640@CELL04-CSE.ENG.UNT.EDU — 134×69

```
-bash: syntax error near unexpected token `newline'
gg0640@cell04-cse:~/assignment$ ./graphGen 100
From:  Moscow, Russia
To  :  Sao Paulo, Brazil
       Riga, Latvia
       Kuwait City, Kuwait
       Geneva, Switzerland
From:  Seoul, South Korea
To  :  Manama, Bahrain
       Seattle, United States
       Abu Dhabi, United Arab Emirates
       Miami, United States
From:  Tokyo, Japan
To  :  Vienna, Austria
       Santiago, Chile
       Riyadh, Saudi Arabia
       Bratislava, Slovakia
       Zurich, Switzerland
       Morristown, United States
       Mexico City, Mexico
From:  Hong Kong, SAR
To  :  Lusaka, Zambia
       Luxembourg, Luxembourg
       Istanbul, Turkey
       San Francisco, United States
       Zagreb, Croatia
       Boston, United States
       Dubai, United Arab Emirates
       New York City, United States
       Houston, United States
       San Juan, Puerto Rico
From:  London, United Kingdom
To  :  Hamburg, Germany
       Washington, United States
       Hong Kong, SAR
       Jakarta, Indonesia
       Barcelona, Spain
       Oslo, Norway
       Lyon, France
From:  Osaka, Japan
To  :  Copenhagen, Denmark
       Dublin, Ireland
       Zagreb, Croatia
From:  Geneva, Switzerland
To  :  Miami, United States
       Budapest, Hungary
       Calgary, Canada
       Cairo, Egypt
       Accra, Ghana
       Abidjan, Cote d'Ivoire
       Berlin, Germany
From:  Copenhagen, Denmark
To  :  Santiago, Chile
       Rome, Italy
       Casablanca, Morocco
       Kiev, Ukraine
       Jeddah, Saudi Arabia
       Birmingham, United Kingdom
       Dakar, Senegal
From:  Oslo, Norway
To  :  Atlanta, United States
       Montreal, Canada
       Copenhagen, Denmark
From:  New York City, United States
To  :  Washington, United States
       Glasgow, United Kingdom
       Milan, Italy
       Guatemala City, Guatemala
       White Plains, United States
```


100%
Sun 28 Apr 10:02 PM
ganesh — gg0640@cell04-cse: ~/assignment — ssh gg0640@CELL04-CSE.ENG.UNT.EDU — 134×69

```
29 -> 23 32 36
30 -> 0
31 ->
32 -> 0 14 16 19 87 90
33 -> 47
34 -> 0 61
35 -> 0 9 34 39 54 60 87
36 -> 40 65
37 -> 43 76 84 89
38 -> 0 48
39 -> 0 7 18 63 66 83
40 -> 0 3 35 36 57
41 -> 0 3 13 21 30 37 59 76
42 ->
43 -> 28 59 69
44 -> 0 11 13 52
45 -> 7 35 40 43 46
46 -> 0 64
47 -> 0 5 25 40
48 -> 0 21 30 32 38 59 66 85
49 -> 68 73
50 -> 0 11 53 58 60 75
51 -> 31
52 ->
53 -> 7
54 -> 0 5 12 16 25 56
55 -> 0 21 46 51 67 80 89
56 -> 0 30 57
57 -> 0 4 8 17 24 34 51
58 -> 0
59 -> 0 49
60 -> 0 3 9 19 29 36 75 86
61 -> 0 55 90
62 -> 0 11 22 31 63 71
63 -> 0 5 16 25
64 ->
65 -> 0 8 60 71 81 83 84
66 -> 9 18 37 41 59
67 ->
68 -> 0 15 29 67 82
69 -> 1
70 -> 0 30 64
71 -> 0 5
72 -> 0 1 33 48 65 75 90
73 -> 0 7 8 61 77 81
74 -> 0 31 61 62
75 -> 43 54
76 -> 0 46 49 51
77 -> 0 33
78 -> 0 44 63 77
79 -> 0 2 4 34 41 46 48
80 -> 0 10 19 57 60 62
81 -> 0 5 16 70 79
82 -> 27 38 60
83 -> 0 23 25 47 65 71
84 -> 0 15 30 59 85 88
85 -> 3 5
86 -> 0 46 48
87 -> 0 3 85
88 -> 0 11 22 36 63 79
89 -> 0 23 78
90 -> 0 66
Please select the type of the questions:
1: From city 'A' to city 'B' with less than x connestions?
2: Route with the samllest number of connections from city 'A' to city 'D' through city 'B' and 'C'?
3: Visit all reachable cities from city 'A' and return to 'A' with the minimum number of connections?
4: Find an good/optimistic meeting city for three friends in different cities?
Enter your choice: incorrect choice
gg0640@cell04-cse:~/assignment$ 
```

Task one :

I am in city “A”, can I fly to city “B” with less than x connections? Give me the route with the smallest number of connections or tell me there is no such a route.

Inputs for this task are as follows .

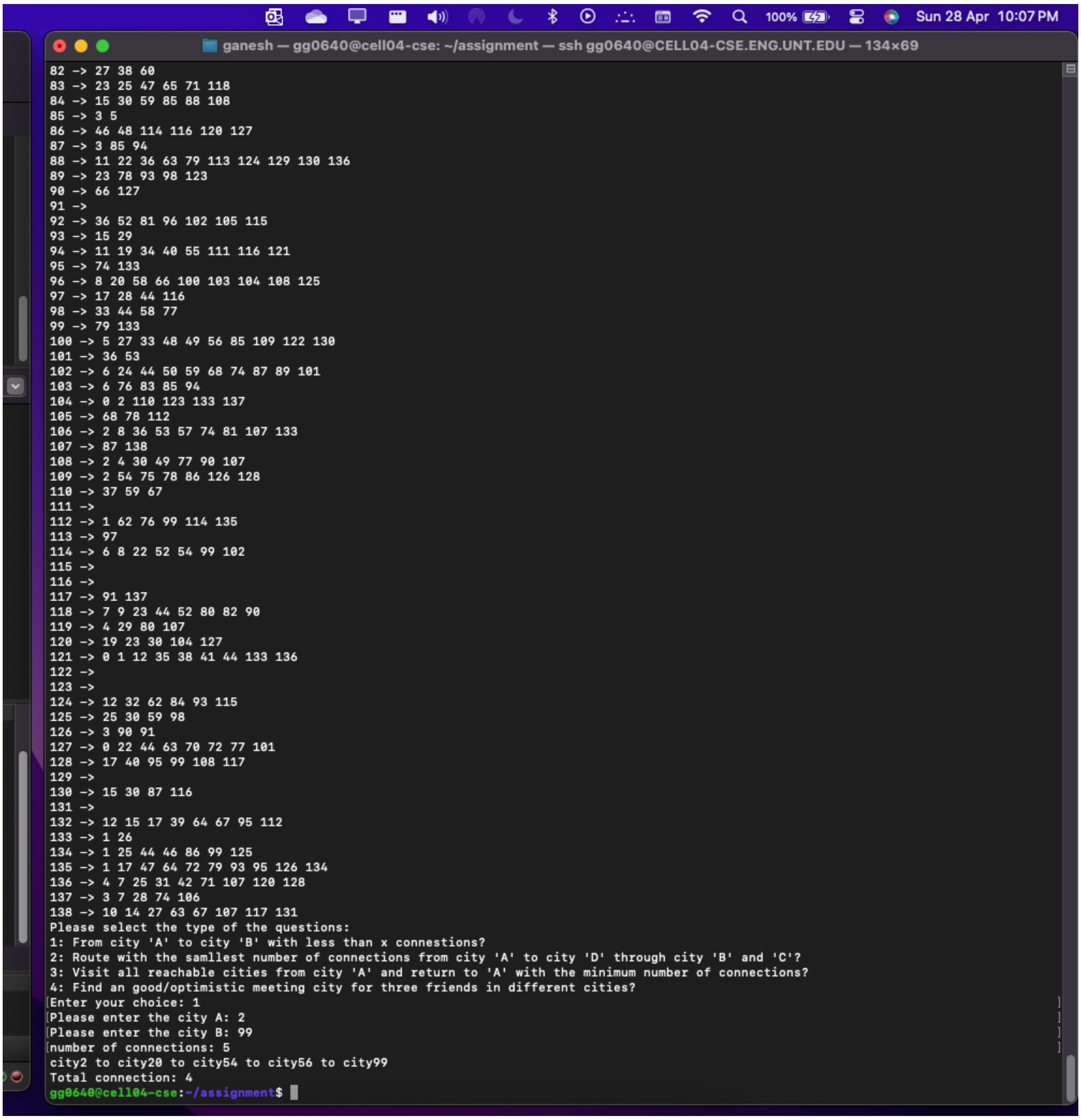
First input :

It is the total no of cities to be considered : 140

Second input :

This is the type of route search we should be performing and it’s a list of choices which requests us , in this case , it is gonna be one.

Inputs : after this , we need to provide the city numbers of one and two and the number of connections that we cannot exceed as being asked by the question .


100%
Sun 28 Apr 10:07 PM
ganesh — gg0640@cell04-cse: ~/assignment — ssh gg0640@CELL04-CSE.ENG.UNT.EDU — 134×69

```
82 -> 27 38 60
83 -> 23 25 47 65 71 118
84 -> 15 30 59 85 88 108
85 -> 3 5
86 -> 46 48 114 116 120 127
87 -> 3 85 94
88 -> 11 22 36 63 79 113 124 129 130 136
89 -> 23 78 93 98 123
90 -> 66 127
91 ->
92 -> 36 52 81 96 102 105 115
93 -> 15 29
94 -> 11 19 34 40 55 111 116 121
95 -> 74 133
96 -> 8 20 58 66 100 103 104 108 125
97 -> 17 28 44 116
98 -> 33 44 58 77
99 -> 79 133
100 -> 5 27 33 48 49 56 85 109 122 130
101 -> 36 53
102 -> 6 24 44 50 59 68 74 87 89 101
103 -> 6 76 83 85 94
104 -> 0 2 110 123 133 137
105 -> 68 78 112
106 -> 2 8 36 53 57 74 81 107 133
107 -> 87 138
108 -> 2 4 30 49 77 90 107
109 -> 2 54 75 78 86 126 128
110 -> 37 59 67
111 ->
112 -> 1 62 76 99 114 135
113 -> 97
114 -> 6 8 22 52 54 99 102
115 ->
116 ->
117 -> 91 137
118 -> 7 9 23 44 52 80 82 90
119 -> 4 29 80 107
120 -> 19 23 30 104 127
121 -> 0 1 12 35 38 41 44 133 136
122 ->
123 ->
124 -> 12 32 62 84 93 115
125 -> 25 30 59 98
126 -> 3 90 91
127 -> 0 22 44 63 70 72 77 101
128 -> 17 40 95 99 108 117
129 ->
130 -> 15 30 87 116
131 ->
132 -> 12 15 17 39 64 67 95 112
133 -> 1 26
134 -> 1 25 44 46 86 99 125
135 -> 1 17 47 64 72 79 93 95 126 134
136 -> 4 7 25 31 42 71 107 120 128
137 -> 3 7 28 74 106
138 -> 10 14 27 63 67 107 117 131
Please select the type of the questions:
1: From city 'A' to city 'B' with less than x connestions?
2: Route with the samllest number of connections from city 'A' to city 'D' through city 'B' and 'C'?
3: Visit all reachable cities from city 'A' and return to 'A' with the minimum number of connections?
4: Find an good/optimistic meeting city for three friends in different cities?
Enter your choice: 1
Please enter the city A: 2
Please enter the city B: 99
number of connections: 5
city2 to city20 to city54 to city56 to city99
Total connection: 4
gg0640@cell04-cse:~/assignment$ 
```

Task two :

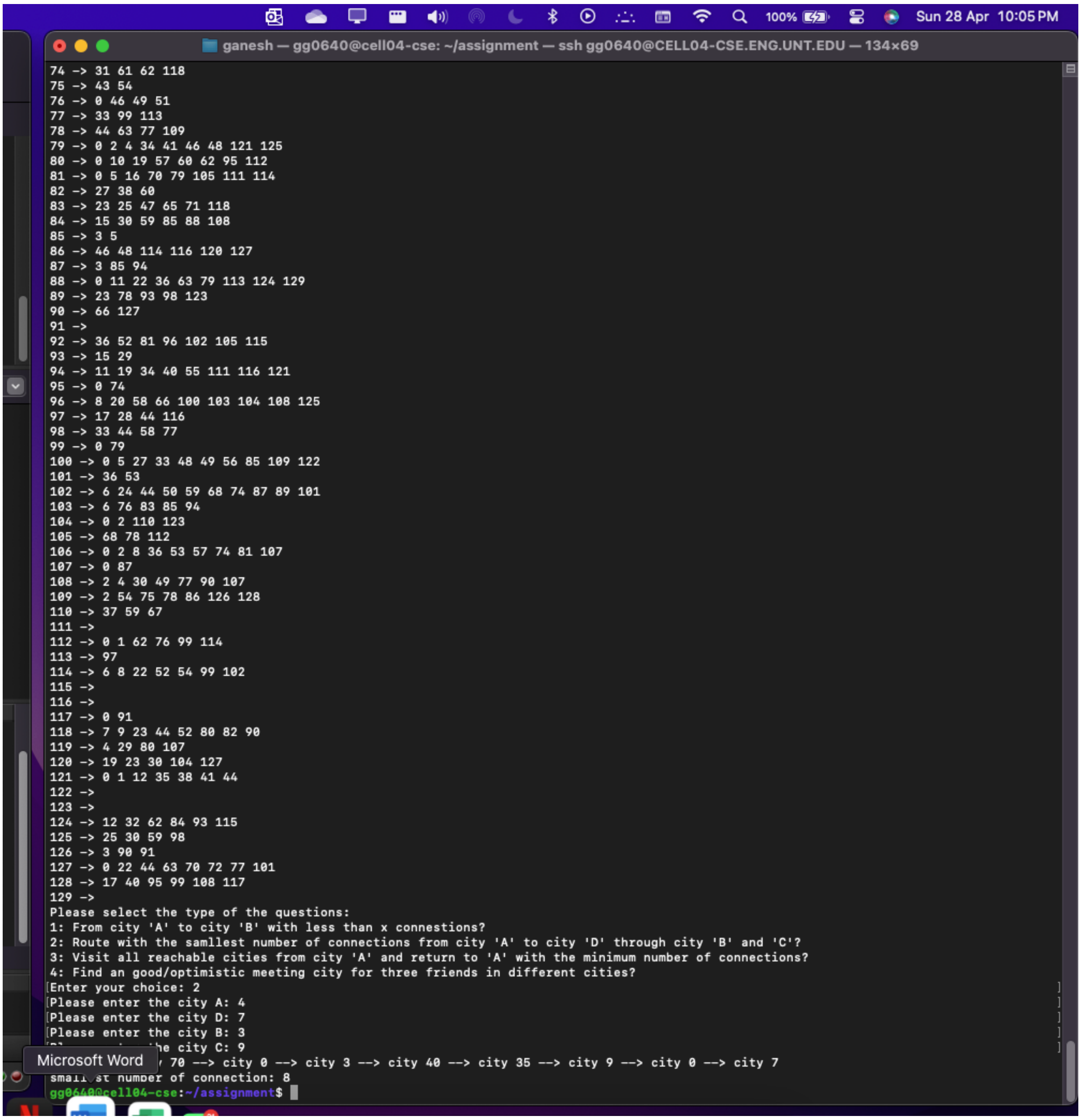
Give me the route with the smallest number of connections from city “A” to city “D” through city “B” and “C”. (the order of “B” and “C” is not important). Or tell me there is no such a route

Sample of inputs :

In the task two , the first and the second inputs are same as the above and for the second one , we should be choosing the choice 2 which implies the task 2 for performing the route search 2 function for this one .

Inputs

Here, the inputs for my case are cites a , b , c, d indices aka numbers all should be given for finding the route through appropriate nodes only .
Output: it gives us on how the cities are reached one by one as shown in the output .


100%
Sun 28 Apr 10:05 PM
ganesh — gg0640@cell04-cse: ~/assignment — ssh gg0640@CELL04-CSE.ENG.UNT.EDU — 134×69
Microsoft Word

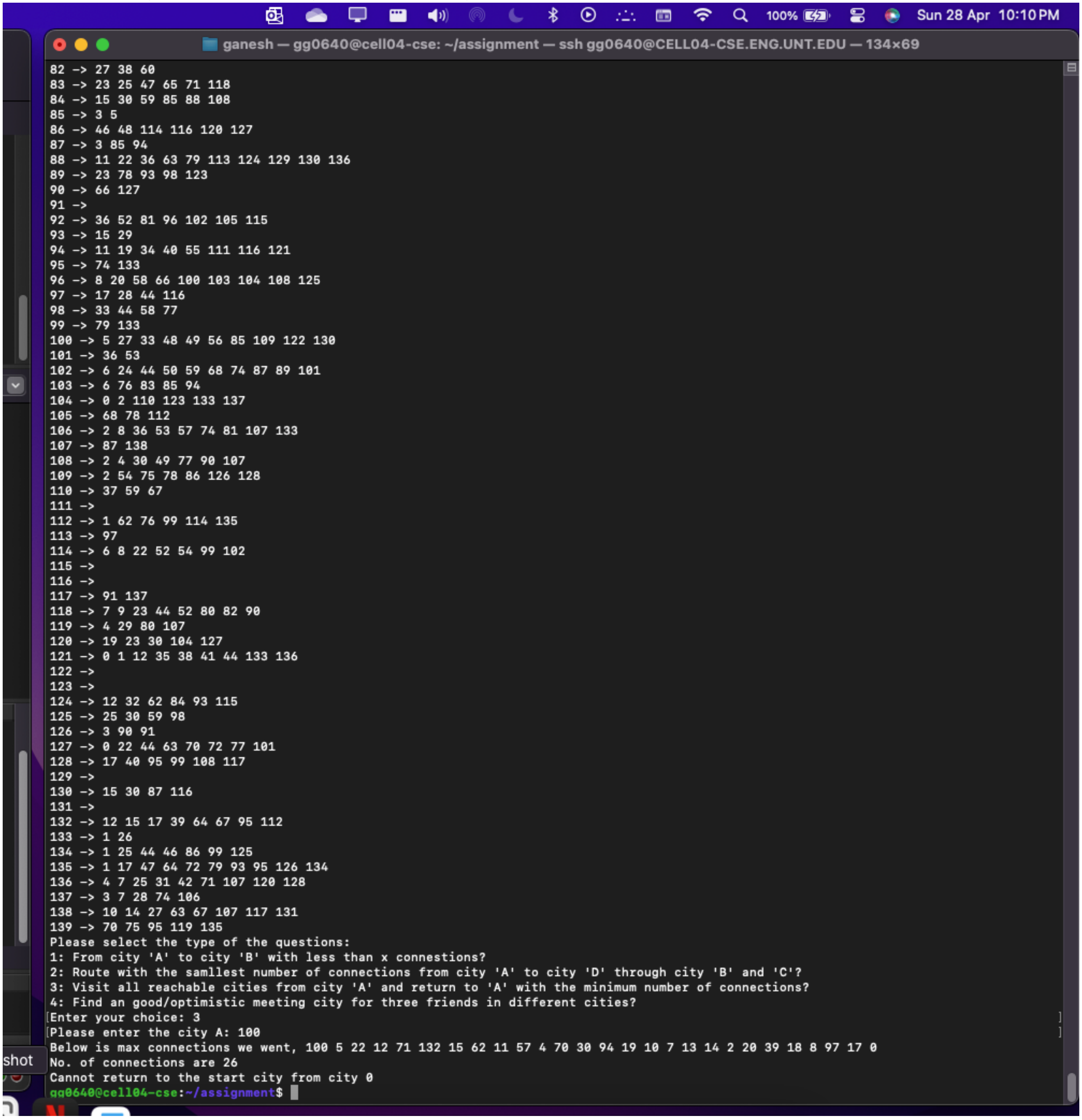
```
74 -> 31 61 62 118
75 -> 43 54
76 -> 0 46 49 51
77 -> 33 99 113
78 -> 44 63 77 109
79 -> 0 2 4 34 41 46 48 121 125
80 -> 0 10 19 57 60 62 95 112
81 -> 0 5 16 70 79 105 111 114
82 -> 27 38 60
83 -> 23 25 47 65 71 118
84 -> 15 30 59 85 88 108
85 -> 3 5
86 -> 46 48 114 116 120 127
87 -> 3 85 94
88 -> 0 11 22 36 63 79 113 124 129
89 -> 23 78 93 98 123
90 -> 66 127
91 ->
92 -> 36 52 81 96 102 105 115
93 -> 15 29
94 -> 11 19 34 40 55 111 116 121
95 -> 0 74
96 -> 8 20 58 66 100 103 104 108 125
97 -> 17 28 44 116
98 -> 33 44 58 77
99 -> 0 79
100 -> 0 5 27 33 48 49 56 85 109 122
101 -> 36 53
102 -> 6 24 44 50 59 68 74 87 89 101
103 -> 6 76 83 85 94
104 -> 0 2 110 123
105 -> 68 78 112
106 -> 0 2 8 36 53 57 74 81 107
107 -> 0 87
108 -> 2 4 30 49 77 90 107
109 -> 2 54 75 78 86 126 128
110 -> 37 59 67
111 ->
112 -> 0 1 62 76 99 114
113 -> 97
114 -> 6 8 22 52 54 99 102
115 ->
116 ->
117 -> 0 91
118 -> 7 9 23 44 52 80 82 90
119 -> 4 29 80 107
120 -> 19 23 30 104 127
121 -> 0 1 12 35 38 41 44
122 ->
123 ->
124 -> 12 32 62 84 93 115
125 -> 25 30 59 98
126 -> 3 90 91
127 -> 0 22 44 63 70 72 77 101
128 -> 17 40 95 99 108 117
129 ->
Please select the type of the questions:
1: From city 'A' to city 'B' with less than x connestions?
2: Route with the samllest number of connections from city 'A' to city 'D' through city 'B' and 'C'?
3: Visit all reachable cities from city 'A' and return to 'A' with the minimum number of connections?
4: Find an good/optimistic meeting city for three friends in different cities?
[Enter your choice: 2
[Please enter the city A: 4
[Please enter the city D: 7
[Please enter the city B: 3
[Pl          he city C: 9
                 70 --> city 0 --> city 3 --> city 40 --> city 35 --> city 9 --> city 0 --> city 7
smal  st number of connection: 8
gg0640@cell04-cse:~/assignment$
```

Task 3 :

I want to start from city “A”, visit all the cities that can be reached and then come back to “A” using as less connections as possible. Give me a route or tell me there is no such a route. (note: once you come back to “A”, you do not go out anymore)

Inputs :
Same for the task one and two are being repeated here .
Here , the choice = 3 is to be selected.

For this one , we need to give the start and the city end index which is the same city number . so essentially , it is telling you to start from a city and return to the same city which is not possible in my case .


100%
Sun 28 Apr 10:10 PM
ganesh — gg0640@cell04-cse: ~/assignment — ssh gg0640@CELL04-CSE.ENG.UNT.EDU — 134×69
shot

```
82 -> 27 38 60
83 -> 23 25 47 65 71 118
84 -> 15 30 59 85 88 108
85 -> 3 5
86 -> 46 48 114 116 120 127
87 -> 3 85 94
88 -> 11 22 36 63 79 113 124 129 130 136
89 -> 23 78 93 98 123
90 -> 66 127
91 ->
92 -> 36 52 81 96 102 105 115
93 -> 15 29
94 -> 11 19 34 40 55 111 116 121
95 -> 74 133
96 -> 8 20 58 66 100 103 104 108 125
97 -> 17 28 44 116
98 -> 33 44 58 77
99 -> 79 133
100 -> 5 27 33 48 49 56 85 109 122 130
101 -> 36 53
102 -> 6 24 44 50 59 68 74 87 89 101
103 -> 6 76 83 85 94
104 -> 0 2 110 123 133 137
105 -> 68 78 112
106 -> 2 8 36 53 57 74 81 107 133
107 -> 87 138
108 -> 2 4 30 49 77 90 107
109 -> 2 54 75 78 86 126 128
110 -> 37 59 67
111 ->
112 -> 1 62 76 99 114 135
113 -> 97
114 -> 6 8 22 52 54 99 102
115 ->
116 ->
117 -> 91 137
118 -> 7 9 23 44 52 80 82 90
119 -> 4 29 80 107
120 -> 19 23 30 104 127
121 -> 0 1 12 35 38 41 44 133 136
122 ->
123 ->
124 -> 12 32 62 84 93 115
125 -> 25 30 59 98
126 -> 3 90 91
127 -> 0 22 44 63 70 72 77 101
128 -> 17 40 95 99 108 117
129 ->
130 -> 15 30 87 116
131 ->
132 -> 12 15 17 39 64 67 95 112
133 -> 1 26
134 -> 1 25 44 46 86 99 125
135 -> 1 17 47 64 72 79 93 95 126 134
136 -> 4 7 25 31 42 71 107 120 128
137 -> 3 7 28 74 106
138 -> 10 14 27 63 67 107 117 131
139 -> 70 75 95 119 135
Please select the type of the questions:
1: From city 'A' to city 'B' with less than x connestions?
2: Route with the samllest number of connections from city 'A' to city 'D' through city 'B' and 'C'?
3: Visit all reachable cities from city 'A' and return to 'A' with the minimum number of connections?
4: Find an good/optimistic meeting city for three friends in different cities?
[Enter your choice: 3
[Please enter the city A: 100
Below is max connections we went, 100 5 22 12 71 132 15 62 11 57 4 70 30 94 19 10 7 13 14 2 20 39 18 8 97 17 0
No. of connections are 26
Cannot return to the start city from city 0
gg0640@cell04-cse:~/assignment$ 
```

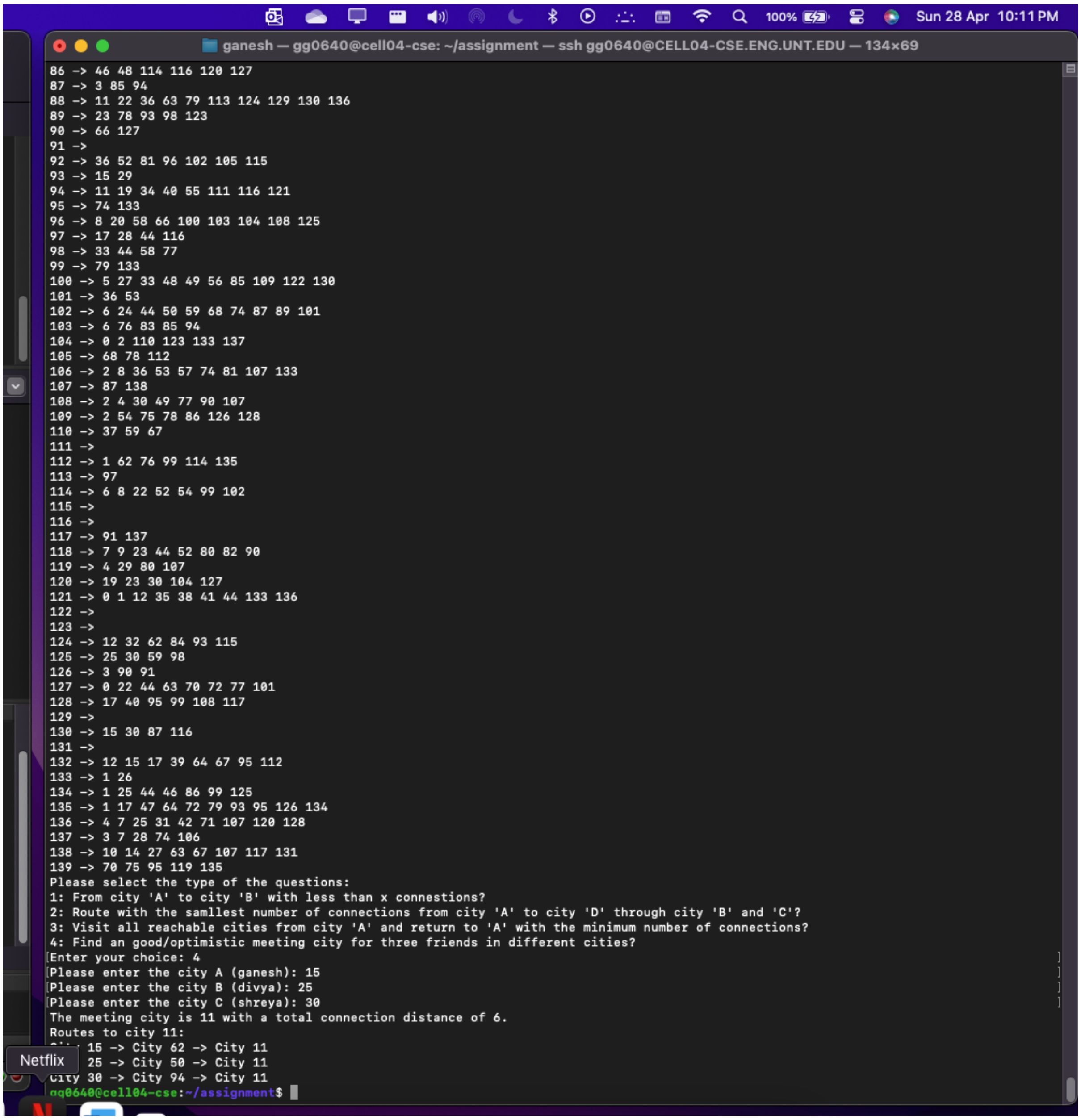
Task four:

Inputs are:

Cities list: 140:

Here , the next inputs are the three cities of our friends , which then selectes a different city to meet so that a minimization of the connections occurs between us three.

Output is given below .



100%
Sun 28 Apr 10:11 PM
ganesh — gg0640@cell04-cse: ~/assignment — ssh gg0640@CELL04-CSE.ENG.UNT.EDU — 134×69
Netflix

```
86 -> 46 48 114 116 120 127
87 -> 3 85 94
88 -> 11 22 36 63 79 113 124 129 130 136
89 -> 23 78 93 98 123
90 -> 66 127
91 ->
92 -> 36 52 81 96 102 105 115
93 -> 15 29
94 -> 11 19 34 40 55 111 116 121
95 -> 74 133
96 -> 8 20 58 66 100 103 104 108 125
97 -> 17 28 44 116
98 -> 33 44 58 77
99 -> 79 133
100 -> 5 27 33 48 49 56 85 109 122 130
101 -> 36 53
102 -> 6 24 44 50 59 68 74 87 89 101
103 -> 6 76 83 85 94
104 -> 0 2 110 123 133 137
105 -> 68 78 112
106 -> 2 8 36 53 57 74 81 107 133
107 -> 87 138
108 -> 2 4 30 49 77 90 107
109 -> 2 54 75 78 86 126 128
110 -> 37 59 67
111 ->
112 -> 1 62 76 99 114 135
113 -> 97
114 -> 6 8 22 52 54 99 102
115 ->
116 ->
117 -> 91 137
118 -> 7 9 23 44 52 80 82 90
119 -> 4 29 80 107
120 -> 19 23 30 104 127
121 -> 0 1 12 35 38 41 44 133 136
122 ->
123 ->
124 -> 12 32 62 84 93 115
125 -> 25 30 59 98
126 -> 3 90 91
127 -> 0 22 44 63 70 72 77 101
128 -> 17 40 95 99 108 117
129 ->
130 -> 15 30 87 116
131 ->
132 -> 12 15 17 39 64 67 95 112
133 -> 1 26
134 -> 1 25 44 46 86 99 125
135 -> 1 17 47 64 72 79 93 95 126 134
136 -> 4 7 25 31 42 71 107 120 128
137 -> 3 7 28 74 106
138 -> 10 14 27 63 67 107 117 131
139 -> 70 75 95 119 135
Please select the type of the questions:
1: From city 'A' to city 'B' with less than x connestions?
2: Route with the samllest number of connections from city 'A' to city 'D' through city 'B' and 'C'?
3: Visit all reachable cities from city 'A' and return to 'A' with the minimum number of connections?
4: Find an good/optimistic meeting city for three friends in different cities?
Enter your choice: 4
Please enter the city A (ganesh): 15
Please enter the city B (divya): 25
Please enter the city C (shreya): 30
The meeting city is 11 with a total connection distance of 6.
Routes to city 11:
City 15 -> City 62 -> City 11
     25 -> City 50 -> City 11
City 30 -> City 94 -> City 11
gg0640@cell04-cse:~/assignment$
```

Part two :

a. Design issues :

1. Here, I found one design issue in the graph class , the number of vertices or nodes and the list of the adjacent vertices are being kept private requiring two functions.
2. Here , coming to the loginc , the design of routesearch() function is to handle the inputs for different tasks and a case structure is being done to choose between differencr tasks .

b. Data structures used :

1. I had used data structures like min priority queue , stack , arrays .
2. Standard c++ library implementations is used for min priority queue and stack .
3. For dijkstra’s algorithm , min priority queue is used.

4. Stack is used for printing the routes.

c. Algorithms used :
1. Dijkstra's algo is used to find shortest paths .
2. Greedy approach in task 3 is done to visit max possible nodes.
3. For other tasks , simple logic is being done .

Running time analysis :

a. Dijkstra's algorithm implementation will be using adjacency matrix and a priority queue in a binary heap , so it has the worst case time complexity of O(n2 logn) and pushing and popping of an element will be taking an O(logn) for a queue .
b. Route search 1 :

   Dijkstra's algo : O(n^2 log n)

   Path is being reconstructed of O(n^2) for insertion of each city in the path for the beginning of the vector .

   Output path is O(n)

   Total time complexity. : O(n^2 log n).

c. Route search 2 :
1. Here , multiple dijkstra's calls are used, the dijkstra's algorithm is basically upto a six times which is dependant on the existence of the paths between the cities .
2. Total time complexity would be six time the O(n^2 log n) which is O(n^2 log n)
3. Path reconstruction is done upto a 6 timers , contributing overall O(n^2)
4. Final steps would be comparing the two integers and iterating it over a list of n elements which contributes lower order terms .
5. Here , the dominating factor in the time complexity for task 2 is the mulpile calls to dijkstra's algorithm each having a time complexity of O(n^2 log n) , the overall is also same .This would be making the function inefficient for very large graphs and if the path involving multiple cities. Optimization would be using more efficient data structures and more route planning .

D. Route search 3 :

Here ,the initial dijkstra's calls , the dijkstra's algorithm is then called once for a each city that is visited until and then all the reachable cities have been visited , in the worst case the graph is then fully connected and every city is then getting reachable from every other algorithm could be invoked upto n times leading to O(n times n^2 log n) = O(n^3 log n) complexity.

Final call would be adding another O(n^2 log n), this will be effecting the overall complexity anyway .

When coming to loop and tour updates , the sum of operations for finding the next city and the updation across all iterations will contribute an extra O(n^2), but this is also being equally dominated by the O(n^3 log n) complexity from the repeated Dijkstra calls which are 6 total .

Dominant factor : repeated invocating of the dijkstra’s algo done upto n time leading upto time complexity of O(n^3 log n) which is inefficient for large graphs .

d. Route search 4 .

Here , in the dijkstra’s algorithm , three calls are being made , and the each call is having a time complexity of O(n^2 log n)

Meeting city cost: O(n) .

Printing routes : assume worst case where the route to the city of meeting could be travelling the entire set of cities and the total time complexity would be for printing all routes is O(N) .

Dominant factor: multiple invoactions of dijkstra’s algorithm .
Overall complexity is : O(n^2 log n).
Primarily due to adjacency matrix making it inefficient .

Scalability:

A. Here, the current implementation of this will be facing some scalability issues with a very large graphs due to its reliance on the dijkstra’s algorithm and the adjacency matrix .
B. Here , the efficiency could be improved by implementation of more efficient data structures and algorithms tailored to the basic characteristics of the graphs .

Remarks

a. Here ,the balancing simplicity and the efficiency of the algo is effectively crucial , especially when dealing with a large scale geographic data .
b. Here , continuous evaluation and optimization is being essential for improving the scalability and the performance .