

MINISTERUL EDUCAȚIEI REPUBLICII MOLDOVA
UNIVERSITATEA TEHNICĂ A MOLDOVEI
Facultatea „Calculatoare, Informatică și Microelectronică”
FILIERA ANGLOFONĂ

REPORT

Lucrare de laborator nr. 1 **la Cercetari Operationale**

A efectuat:

st. gr. FAF-151

Tanașciuc Macarie

A verificat:

Victor Țurcanu

Chisinau 2017

Problem description

Aunt Marry is an old lady from Chişinău. She lives in the area near the Circus and everyday she needs to take the trolleybus in order to get to the city center. From there she can go further to the *Piaţa Centrală*, where she doesn't buy anything, rather she spends her time hearing the latest news in the city.

Unfortunately, Aunt Marry is upset by the way the municipal transport is optimized especially when she returns back home. For example, from station **Ştefan cel Mare** to station **Circul** there go trolleybuses nr: 7, 10, 24, 25 (I will skip 11 and 14 as they are routed comparably few times per day). Thus, she doesn't care so much what is the number of the trolleybus she takes in order to reach the destination station. One thing she noticed is that often the trolleybuses arrive as a train (all at once) to the station, but there appear gaps of around 10 or even 15 minutes when there no trolleybuses arrive. What Aunt Marry wants, neglecting the car traffic in our city, is an evenly distributed timetable when she can wait for the minimum amount of time the transportation that will take her to the destination.

[Here](#) is represented the mini graph of Chisinau's public transportation.

Subproblem 1

Given the [timetables](#) of trolleybus arrival, find an algorithm to optimize the traffic from station **Ştefan cel Mare** to station **Circul**. Visualize the data output of your solution.

Subproblem 2

Given the [timetables](#) of trolleybus arrival, find an algorithm to optimize the traffic among all given stations.

Subproblem 3

Given the [passenger traffic load](#) between two stations (written in the parentheses), find an algorithm to determine the number of trolleybuses of each type needed to satisfy the law obedient citizens' needs that use the public transportation. For example, how many trolleybuses of number 3 are needed to completely consume the load? If there isn't sufficient data, what would you need more to take into consideration?

PS: Consider that there can be max 70 people in a trolleybus.

[illegible]

-----Time table asem -----

7) 11:35, 11:53, 12:11, 12:29, 12:47, 13:05, 13:23

10) 11:40, 11:55, 12:09, 12:19, 12:27, 12:37, 12:50, 13:08, 13:22

24) 11:41, 11:56, 12:09, 12:19, 12:26, 12:36, 12:48, 13:06, 13:20

25) 11:34, 11:52, 12:10, 12:28, 12:46, 13:04, 13:22

•Index all: [0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 3, 3]

•Time all: [35, 53, 71, 89, 107, 125, 143, 40, 55, 69, 79, 87, 97, 110, 128, 142, 41, 56, 69, 79, 86, 96, 108, 126, 140, 34, 52, 70, 88, 106, 124, 142]

○Index all sorted: [3, 0, 1, 2, 3, 0, 1, 2, 1, 2, 3, 0, 1, 2, 2, 1, 3, 0, 2, 1, 3, 0, 2, 1, 3, 0, 2, 1, 2, 1, 3, 0]

○Time all sorted [34, 35, 40, 41, 52, 53, 55, 56, 69, 69, 70, 71, 79, 79, 86, 87, 88, 89, 96, 97, 106, 107, 108, 110, 124, 125, 126, 128, 140, 142, 142, 143]

GAP before= [1, 5, 1, 11, 1, 2, 1, 13, 0, 1, 1, 8, 0, 7, 1, 1, 1, 7, 1, 9, 1, 1, 2, 14, 1, 1, 2, 12, 2, 0, 1]

- Total Gap before 109

- Average gap before= 3.5161290322580645

GAP after= [3, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 4]

- Total gap after= 108

- Average gap after= 3.4838709677419355

[30, 32, 36, 37, 48, 49, 51, 52, 65, 65, 66, 67, 75, 75, 82, 83, 85, 86, 93, 94, 103, 104, 105, 107, 121, 122, 123, 125, 137, 139, 139, 139]

Minutes from 11:00

[[32, 49, 67, 86, 104, 122, 139],

[36, 51, 65, 75, 83, 94, 107, 125, 139],

[37, 52, 65, 75, 82, 93, 105, 123, 137],

[30, 48, 66, 85, 103, 121, 139]]

Optimized time table

[['7) 11:32 11:49 12:07 12:26 12:44 13:02 13:19'],

[10) 11:36 11:51 12:05 12:15 12:23 12:34 12:47 13:05 13:19].

[24) 11:37 11:52 12:05 12:15 12:22 12:33 12:45 13:03 13:17],

['25) 11:30 11:48 12:06 12:25 12:43 13:01 13:19']]

[illegible]

-----Time table casa_presei -----

2) 11:32, 11:50, 12:05, 12:15, 12:24, 12:33, 12:45, 13:01, 13:15

10) 11:33, 11:48, 12:02, 12:12, 12:20, 12:30, 12:43, 13:01, 13:15

24) 11:34, 11:49, 12:02, 12:12, 12:19, 12:29, 12:41, 12:59, 13:13

- Index all: [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2]

[45, 47, 61, 62, 75, 74, 85, 84, 92, 93, 102, 103, 114, 116, 132, 134, 146, 147]

[[45, 61, 75, 85, 93, 103, 116, 134, 147],
[47, 62, 74, 84, 92, 102, 114, 132, 146]]

[['10) 11:45 12:01 12:15 12:25 12:33 12:43 12:56 13:14 13:27'],
['24) 11:47 12:02 12:14 12:24 12:32 12:42 12:54 13:12 13:26']]

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

2) 11:33, 11:48, 12:03, 12:13, 12:22, 12:31, 12:43, 12:59, 13:13
3) 11:34, 11:50, 12:05, 12:16, 12:31, 12:46, 13:03
10) 11:31, 11:46, 12:00, 12:10, 12:18, 12:28, 12:41, 12:59, 13:13
24) 11:32, 11:47, 12:00, 12:10, 12:17, 12:27, 12:39, 12:57, 13:11

=====

•Time all: [33, 48, 63, 73, 82, 91, 103, 119, 133, 34, 50, 65, 76, 91, 106, 123, 31, 46, 60, 70, 78, 88, 101, 119, 133, 32, 47, 60, 70, 77, 87, 99, 117, 131]

○Time all sorted [31, 32, 33, 34, 46, 47, 48, 50, 60, 60, 63, 65, 70, 70, 73, 76, 77, 78, 82, 87, 88, 91, 91, 99, 101, 103, 106, 117, 119, 119, 123, 131, 133, 133]

- Average gap before= 3.090909090909091

- Average gap after= 3.0606060606060606

[30, 45, 60, 70, 79, 88, 100, 116, 130],
 [31, 47, 62, 73, 88, 103, 119],
 [27, 43, 57, 67, 75, 85, 98, 116, 129],
 [29, 44, 57, 67, 74, 84, 96, 114, 128]]

[['2) 11:30 11:45 12:00 12:10 12:19 12:28 12:40 12:56 13:10'],
['3) 11:31 11:47 12:02 12:13 12:28 12:43 12:59'],
['10) 11:27 11:43 11:57 12:07 12:15 12:25 12:38 12:56 13:09'],
['24) 11:29 11:44 11:57 12:07 12:14 12:24 12:36 12:54 13:08']]

[illegible]

7) 11:28, 11:46, 12:04, 12:22, 12:40, 12:58, 13:16
25) 11:27, 11:45, 12:03, 12:21, 12:39, 12:57, 13:16

GAP after= [8, 8, 8, 8, 8, 9, 9, 8, 9, 8, 9, 8, 8]
•Total gap after= 108
•Average gap after= 8.307692307692308
[19, 20, 37, 38, 55, 56, 72, 73, 91, 91, 109, 109, 128, 128]
Minutes from 11:00
[[20, 38, 56, 73, 91, 109, 128], [19, 37, 55, 72, 91, 109, 128]]
Optimized time table
[['(7) 11:20 11:38 11:56 12:13 12:31 12:49 13:08'],
['(25) 11:19 11:37 11:55 12:12 12:31 12:49 13:08']]

-----Time table mihai eminescu theatre -----

GAP after= [17, 15, 10, 9, 9, 12, 16, 14]
 •Total gap after= 102
 •Average gap after= 12.75
 [21, 36, 53, 68, 78, 87, 96, 108, 124]
 Minutes from 11:00
 [[21, 36, 53, 68, 78, 87, 96, 108, 124]]
 Optimized time table
 [[('2) 11:21 11:36 11:53 12:08 12:18 12:27 12:36 12:48 13:04']]

To solve the second problem I applied the same algorithm to the 1st problem.

Solution problem 3

Route 2 needs 9 trolleybuses

Route 3 needs 7 trolleybuses

Route 7 needs 6 trolleybuses

Route 10 needs 10 trolleybuses

Route 24 needs 10 trolleybuses

Route 25 needs 9 trolleybuses