MINISTERUL EDUCAŢIEI REPUBLICII MOLDOVA

UNIVERSITATEA TEHNICĂ A MOLDOVEI

Facultatea „Calculatoare, Informatică şi 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 Chișinău. 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 *Piața 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 **Ștefan 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 **Ștefan 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.

Solution for problem 1-2

■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■

----------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, 1, 1, 1, 1, 1, 1, 1, 1, 1, 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, 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']]

■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■

----------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, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2]

•Time all: [32, 50, 65, 75, 84, 93, 105, 121, 135, 33, 48, 62, 72, 80, 90, 103, 121, 135, 34, 49, 62, 72, 79, 89, 101, 119, 133]
○Index all sorted: [0, 1, 2, 1, 2, 0, 1, 2, 0, 1, 2, 0, 2, 1, 0, 2, 1, 0, 2, 1, 0, 2, 0, 1, 2, 0, 1]
○Time all sorted [32, 33, 34, 48, 49, 50, 62, 62, 65, 72, 72, 75, 79, 80, 84, 89, 90, 93, 101, 103, 105, 119, 121, 121, 133, 135, 135]
GAP before= [1, 1, 14, 1, 1, 12, 0, 3, 7, 0, 3, 4, 1, 4, 5, 1, 3, 8, 2, 2, 14, 2, 0, 12, 2, 0]
•Total Gap before 103
•Average gap before= 3.9615384615384617
==========
GAP after= [4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4]
•Total gap after= 104
•Average gap after= 4.0
[28, 29, 30, 44, 45, 46, 58, 58, 61, 68, 68, 71, 75, 76, 80, 85, 86, 89, 97, 99, 101, 115, 117, 117, 129, 131, 131]
Minutes from 11:00
[[28, 46, 61, 71, 80, 89, 101, 117, 131],
 [29, 44, 58, 68, 76, 86, 99, 117, 131],
 [30, 45, 58, 68, 75, 85, 97, 115, 129]]
Optimized time table
[['2) 11:28 11:46 12:01 12:11 12:20 12:29 12:41 12:57 13:11'],
 ['10) 11:29 11:44 11:58 12:08 12:16 12:26 12:39 12:57 13:11'],
 ['24) 11:30 11:45 11:58 12:08 12:15 12:25 12:37 12:55 13:09']]

■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■

----------Time table central_typography ----------
25) 11:45, 12:03, 12:21, 12:39, 12:57, 13:15, 13:33

==========
•Index all: [0, 0, 0, 0, 0, 0, 0]
•Time all: [45, 63, 81, 99, 117, 135, 153]
○Index all sorted: [0, 0, 0, 0, 0, 0, 0]
○Time all sorted [45, 63, 81, 99, 117, 135, 153]
GAP before= [18, 18, 18, 18, 18, 18]
•Total Gap before 108
•Average gap before= 18.0
==========
GAP after= [17, 18, 18, 18, 18, 18]
•Total gap after= 107
•Average gap after= 17.833333333333332
[27, 46, 63, 81, 99, 117, 135]
Minutes from 11:00
[[27, 46, 63, 81, 99, 117, 135]]
Optimized time table
[['25) 11:27 11:46 12:03 12:21 12:39 12:57 13:15']]

■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■

----------Time table circul ----------
7) 11:42, 12:00, 12:18, 12:36, 12:54, 13:12, 13:30
10) 11:47, 12:02, 12:16, 12:26, 12:34, 12:44, 12:57, 13:15, 13:29
24) 11:48, 12:03, 12:16, 12:26, 12:33, 12:43, 12:55, 13:13, 13:27
25) 11:41, 11:59, 12:17, 12:35, 12:53, 13:11, 13:29

==========
•Index all: [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: [42, 60, 78, 96, 114, 132, 150, 47, 62, 76, 86, 94, 104, 117, 135, 149, 48, 63, 76, 86, 93, 103, 115, 133, 147, 41, 59, 77, 95, 113, 131, 149]
○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 [41, 42, 47, 48, 59, 60, 62, 63, 76, 76, 77, 78, 86, 86, 93, 94, 95, 96, 103, 104, 113, 114, 115, 117, 131, 132, 133, 135, 147, 149, 149, 150]
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, 4]
•Total gap after= 108
•Average gap after= 3.4838709677419355
[37, 39, 43, 44, 55, 56, 58, 59, 72, 72, 73, 74, 82, 82, 89, 90, 92, 93, 100, 101, 110, 111, 112, 114, 128, 129, 130, 132, 144, 146, 146, 146]
Minutes from 11:00
[[39, 56, 74, 93, 111, 129, 146],
 [43, 58, 72, 82, 90, 101, 114, 132, 146],
 [44, 59, 72, 82, 89, 100, 112, 130, 144],
 [37, 55, 73, 92, 110, 128, 146]]
Optimized time table
[['7) 11:39 11:56 12:14 12:33 12:51 13:09 13:26'],
 ['10) 11:43 11:58 12:12 12:22 12:30 12:41 12:54 13:12 13:26'],
 ['24) 11:44 11:59 12:12 12:22 12:29 12:40 12:52 13:10 13:24'],
 ['25) 11:37 11:55 12:13 12:32 12:50 13:08 13:26']]

■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■

----------Time table eminescu ----------
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, 13:00
10) 11:28, 11:43, 11:57, 12:07, 12:15, 12:25, 12:38, 12:56, 13:10
24) 11:29, 11:44, 11:57, 12:07, 12:14, 12:24, 12:36, 12:54, 13:08

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

•Time all: [30, 45, 60, 70, 79, 88, 100, 116, 130, 31, 47, 62, 73, 88, 103, 120, 28, 43, 57, 67, 75, 85, 98, 116, 130, 29, 44, 57, 67, 74, 84, 96, 114, 128]
○Index all sorted: [2, 3, 0, 1, 2, 3, 0, 1, 2, 3, 0, 1, 2, 3, 0, 1, 3, 2, 0, 3, 2, 0, 1, 3, 2, 0, 1, 3, 0, 2, 1, 3, 0, 2]
○Time all sorted [28, 29, 30, 31, 43, 44, 45, 47, 57, 57, 60, 62, 67, 67, 70, 73, 74, 75, 79, 84, 85, 88, 88, 96, 98, 100, 103, 114, 116, 116, 120, 128, 130, 130]
GAP before= [1, 1, 1, 12, 1, 1, 2, 10, 0, 3, 2, 5, 0, 3, 3, 1, 1, 4, 5, 1, 3, 0, 8, 2, 2, 3, 11, 2, 0, 4, 8, 2, 0]
•Total Gap before 102
•Average gap before= 3.090909090909091
==========
GAP after= [3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 4, 3, 3, 4]
•Total gap after= 101
•Average gap after= 3.0606060606060606
[24, 26, 27, 28, 40, 41, 42, 44, 54, 54, 57, 59, 64, 64, 67, 70, 71, 72, 76, 81, 82, 85, 85, 93, 95, 97, 100, 111, 113, 113, 116, 125, 127, 126]
Minutes from 11:00
[[27, 42, 57, 67, 76, 85, 97, 113, 127],
 [28, 44, 59, 70, 85, 100, 116],
 [24, 40, 54, 64, 72, 82, 95, 113, 126],
 [26, 41, 54, 64, 71, 81, 93, 111, 125]]
Optimized time table
[['2) 11:27 11:42 11:57 12:07 12:16 12:25 12:37 12:53 13:07'],
 ['3) 11:28 11:44 11:59 12:10 12:25 12:40 12:56'],
 ['10) 11:24 11:40 11:54 12:04 12:12 12:22 12:35 12:53 13:06'],
 ['24) 11:26 11:41 11:54 12:04 12:11 12:21 12:33 12:51 13:05']]

■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■

----------Time table kiev ----------
10) 11:51, 12:06, 12:20, 12:30, 12:38, 12:48, 13:01, 13:19, 13:33
24) 11:52, 12:07, 12:20, 12:30, 12:37, 12:47, 12:59, 13:17, 13:31

==========
•Index all: [0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1]
•Time all: [51, 66, 80, 90, 98, 108, 121, 139, 153, 52, 67, 80, 90, 97, 107, 119, 137, 151]
○Index all sorted: [0, 1, 0, 1, 0, 1, 0, 1, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0]
○Time all sorted [51, 52, 66, 67, 80, 80, 90, 90, 97, 98, 107, 108, 119, 121, 137, 139, 151, 153]
GAP before= [1, 14, 1, 13, 0, 10, 0, 7, 1, 9, 1, 11, 2, 16, 2, 12, 2]
•Total Gap before 102
•Average gap before= 6.0
==========
GAP after= [5, 5, 5, 5, 6, 5, 6, 5, 5, 5, 5, 5, 5, 5, 5, 5, 6]
•Total gap after= 88
•Average gap after= 5.176470588235294
[45, 47, 61, 62, 75, 74, 85, 84, 92, 93, 102, 103, 114, 116, 132, 134, 146, 147]

Minutes from 11:00
[[45, 61, 75, 85, 93, 103, 116, 134, 147],
 [47, 62, 74, 84, 92, 102, 114, 132, 146]]
Optimized time table
[['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']]

■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■

----------Time table kogalniceanu ----------
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

==========
•Index all: [0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 3, 3, 3, 3]
•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]
○Index all sorted: [2, 3, 0, 1, 2, 3, 0, 1, 2, 3, 0, 1, 2, 3, 0, 1, 3, 2, 0, 3, 2, 0, 1, 3, 2, 0, 1, 3, 0, 2, 1, 3, 0, 2]
○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]
GAP before= [1, 1, 1, 12, 1, 1, 2, 10, 0, 3, 2, 5, 0, 3, 3, 1, 1, 4, 5, 1, 3, 0, 8, 2, 2, 3, 11, 2, 0, 4, 8, 2, 0]
•Total Gap before 102
•Average gap before= 3.090909090909091
==========
GAP after= [3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 4, 3, 3, 4]
•Total gap after= 101
•Average gap after= 3.0606060606060606
[27, 29, 30, 31, 43, 44, 45, 47, 57, 57, 60, 62, 67, 67, 70, 73, 74, 75, 79, 84, 85, 88, 88, 96, 98, 100, 103, 114, 116, 116, 119, 128, 130, 129]
Minutes from 11:00
[[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]]
Optimized time table
[['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']]

■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■

----------Time table licurici ----------
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

==========
•Index all: [0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1]
•Time all: [28, 46, 64, 82, 100, 118, 136, 27, 45, 63, 81, 99, 117, 136]
○Index all sorted: [1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 0, 1]
○Time all sorted [27, 28, 45, 46, 63, 64, 81, 82, 99, 100, 117, 118, 136, 136]
GAP before= [1, 17, 1, 17, 1, 17, 1, 17, 1, 17, 1, 18, 0]
•Total Gap before 109
•Average gap before= 8.384615384615385
==========
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 ----------
2) 11:35, 11:53, 12:08, 12:18, 12:27, 12:36, 12:48, 13:04, 13:18

==========
•Index all: [0, 0, 0, 0, 0, 0, 0, 0, 0]
•Time all: [35, 53, 68, 78, 87, 96, 108, 124, 138]
○Index all sorted: [0, 0, 0, 0, 0, 0, 0, 0, 0]
○Time all sorted [35, 53, 68, 78, 87, 96, 108, 124, 138]
GAP before= [18, 15, 10, 9, 9, 12, 16, 14]
•Total Gap before 103
•Average gap before= 12.875
==========
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']]

■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■

----------Time table puskin ----------
3) 11:36, 11:52, 12:07, 12:18, 12:33, 12:48, 13:05

==========
•Index all: [0, 0, 0, 0, 0, 0, 0]
•Time all: [36, 52, 67, 78, 93, 108, 125]
○Index all sorted: [0, 0, 0, 0, 0, 0, 0]
○Time all sorted [36, 52, 67, 78, 93, 108, 125]
GAP before= [16, 15, 11, 15, 15, 17]
•Total Gap before 89
•Average gap before= 14.833333333333334
==========
GAP after= [16, 15, 11, 15, 15, 16]
•Total gap after= 88
•Average gap after= 14.666666666666666
[20, 36, 52, 67, 78, 93, 109]
Minutes from 11:00
[[20, 36, 52, 67, 78, 93, 109]]
Optimized time table
[['3) 11:20 11:36 11:52 12:07 12:18 12:33 12:49']]

■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■

----------Time table stefan_cel_mare ----------
7) 11:32, 11:50, 12:08, 12:26, 12:44, 13:02, 13:20
10) 11:37, 11:52, 12:06, 12:16, 12:24, 12:34, 12:47, 13:05, 13:19
24) 11:38, 11:53, 12:06, 12:16, 12:23, 12:33, 12:45, 13:03, 13:17
25) 11:31, 11:49, 12:07, 12:25, 12:43, 13:01, 13:19

==========
•Index all: [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: [32, 50, 68, 86, 104, 122, 140, 37, 52, 66, 76, 84, 94, 107, 125, 139, 38, 53, 66, 76, 83, 93, 105, 123, 137, 31, 49, 67, 85, 103, 121, 139]
○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 [31, 32, 37, 38, 49, 50, 52, 53, 66, 66, 67, 68, 76, 76, 83, 84, 85, 86, 93, 94, 103, 104, 105, 107, 121, 122, 123, 125, 137, 139, 139, 140]
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, 4]
•Total gap after= 108
•Average gap after= 3.4838709677419355

[27, 29, 33, 34, 45, 46, 48, 49, 62, 62, 63, 64, 72, 72, 79, 80, 82, 83, 90, 91, 100, 101, 102, 104, 118, 119, 120, 122, 134, 136, 136, 136]
Minutes from 11:00
[[29, 46, 64, 83, 101, 119, 136],
 [33, 48, 62, 72, 80, 91, 104, 122, 136],
 [34, 49, 62, 72, 79, 90, 102, 120, 134],
 [27, 45, 63, 82, 100, 118, 136]]
Optimized time table
[['7) 11:29 11:46 12:04 12:23 12:41 12:59 13:16'],
 ['10) 11:33 11:48 12:02 12:12 12:20 12:31 12:44 13:02 13:16'],
 ['24) 11:34 11:49 12:02 12:12 12:19 12:30 12:42 13:00 13:14'],
 ['25) 11:27 11:45 12:03 12:22 12:40 12:58 13:16']]

■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■

----------Time table vladimirescu ----------
7) 11:45, 12:03, 12:21, 12:39, 12:57, 13:15, 13:33

==========
•Index all: [0, 0, 0, 0, 0, 0, 0]
•Time all: [45, 63, 81, 99, 117, 135, 153]
○Index all sorted: [0, 0, 0, 0, 0, 0, 0]
○Time all sorted [45, 63, 81, 99, 117, 135, 153]
GAP before= [18, 18, 18, 18, 18, 18]
•Total Gap before 108
•Average gap before= 18.0
==========
GAP after= [17, 18, 18, 18, 18, 18]
•Total gap after= 107
•Average gap after= 17.833333333333332
[27, 46, 63, 81, 99, 117, 135]
Minutes from 11:00
[[27, 46, 63, 81, 99, 117, 135]]
Optimized time table
[['7) 11:27 11:46 12:03 12:21 12:39 12:57 13:15']]

Solution:

To solve the 1st problem transformed all time for all intervals of trolley then sorted them and calculated the sum and average time needed to wait and if the gap was smaller than the average I added the average if it was bigger I subtracted it.
To solve the second problem I applied the same algorithm to the 1st problem.

Solution problem 3

Route 2 needs 9 troleybuses
Route 3 needs 7 troleybuses
Route 7 needs 6 troleybuses
Route 10 needs 10 troleybuses
Route 24 needs 10 troleybuses
Route 25 needs 9 troleybuses

Solution:
To solve the 1st problem transformed all time for all intervals of trolley then sorted them and calculated the sum and average time needed to wait and if the gap was smaller than the average I added the average if it was bigger I subtracted it.