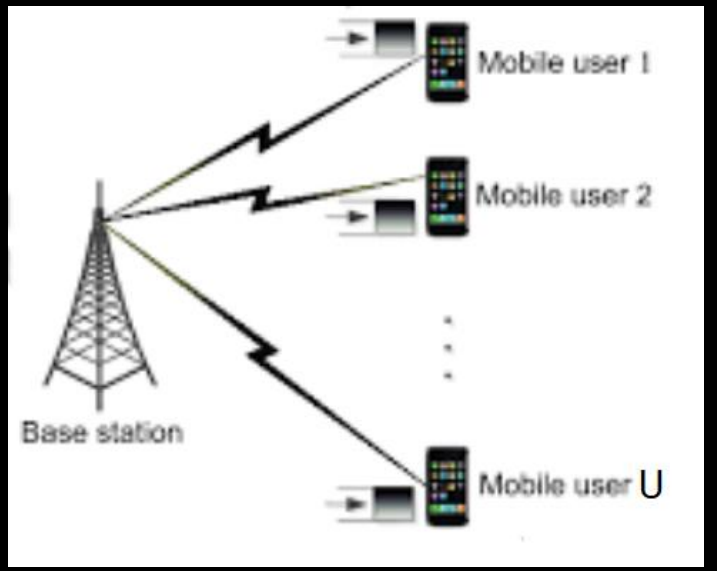
Hackathon Task Description

Background:
Downlink channel allocation problem

- Many users connected to a base station need to send and receive data in real-time.
- As we have limited bandwidth, efficient allocation of data channel resources to the users has a significant impact on users' quality of experience.
- Here, we focus on the simplified version of the downlink data channel allocation problem as a key building block of mobile networks.



Problem statement

- Let’s make the considered problem one step closer to the original problem in real-world by adding another factor to the problem which is users’ weight.
- Indeed, it is the same problem as the qualification task, the only difference is that users have different weights, in the sense that the speed of users with higher weights is more important than users with lower weights.
- The users’ importance is shown by a wieght, where W_i shows the importance of U_i.

Score Function

- Objective: Maximise the weighted sum of average speed of all users (Avg_Speed_i)

$$\text{Objective\_function} = \frac{\sum_{\forall\ Users} Avg\_Speed_i * W_i}{BestSpeedUsers}$$

BestSpeedUsers is the wiegted sum of maximum speed of all users and can be calculated by $\sum_{\forall\ Users} Init\_Speed_i * W_i$

- The same constraint and penalty term as qualification task are applied.
- The score is formulated as:

$$\text{Go}al\text{\_function} = \text{Objective\_function} - \alpha * Penalty\_term$$

Limitations (exactly the same as the qualification phase)

1. Write a code in C++ or Python that can find a placement for all the test cases,
2. The execution time of the code, for each test case, should be less than 1 second on your own machine. If the execution time is greater than one second, the solution is not counted as a valid solution.
3. Only standard libraries can be used, using optimization libraries/tools are not allowed.

Input and Output Data

- Input:
  a. A speed to data map as a csv file, used for all test cases
  b. A set of input files each of which corresponds to a test case where each test case includes
     1. Grid size (M, N)
     2. Number of users (|U|)
     3. Value of α
     4. Users' information (initial speed, data size, factor, weight)
- Submission file (exactly the same as qualification phase): It should be a zip file including:
  a. A csv file for each test case that includes:
     1. Grid placement
     2. Penalty_term
     3. Objective_function
     4. Score
     5. Execution time of the code
  b. Single source file with either .py or .cpp or .cc extension

Output CSV Template (exactly the same as the qualification phase)

N

M

0	U1	U1	U1	U2	-		
1	U2	-	U2	U3	-		
2	U3	-	U4	U4	-		
3	U6	-	-	U5	-		
4	-	-	-	U6	-		
...	-	-	-	-	-		
M-1	-	-	-	-	-		
M	$Data\_Loss_1$	$Data\_Loss_2$	$Data\_Loss_3$	$Data\_Loss_4$	$Data\_Loss_5$	$Data\_Loss_6$	Penalty_term
M+1	Avg_Speed_1	Avg_Speed_2	Avg_Speed_3	Avg_Speed_4	Avg_Speed_5	Avg_Speed_6	Objective_function
M+2	Score						
M+3	Exe Time (msec)						

Testcases

- For the Hackathon phase, there are 10 test cases, some of them derived from the qualification phase.
- *The following is an example input testcase file represented by .csv format:*
- 2,5 *-> Input grid size (Number of Rows (M) and Columns (N), respectively)*
- 4 *-> Total number of users*
- 100 *-> Value of α*
- 1 21 4000 0.85 1 *-> Users' information (User ID, Initial speed, Data size, Factor, weight)*
- 2 18 2800 0.75 2
- 3 19 3900 0.05 1
- 4 25 4500 0.5 3

Evaluation and ranking

Evaluation for the Hackathon includes two main parts:

1. 60% based on the leaderboard, the first team score is considered as 60% and other teams' scores will be scaled accordingly

   Example: If total score of team1, team2 and team3 are 8.5, 6.1, 4.2 respectively, then team1 that has the highest score achives 60%, team2 achives 43%, and team3 achives 29.6%.

2. 40% based on the jury committe's score who asses each team solution during the pitch session

Note1: The jury will evaluate the result based on:

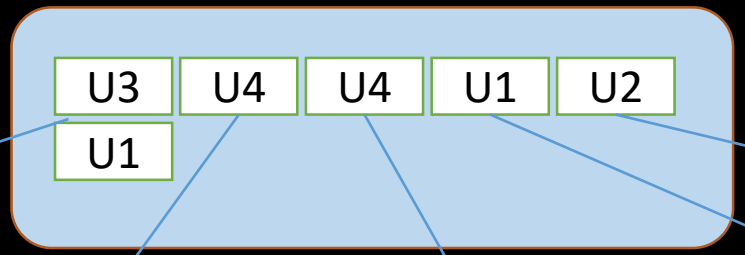
1. Creativity and Innovation of the proposed solution (15%)
2. Presentation skills of your team to clearly describe the idea and analysis (15%)
3. Applicability of the proposed solution in the real world, such as scalability and performance (10%)

Note 2: (The leaderboard ranking is exactly the same as qualification phase)

Example 1

- U1={20, 5000, 0.3, 1}, U2={15, 3500, 0.25, 2}, U3={26, 4200, 0.6, 1}, U4={20, 10000, 0.2 , 3}; α = 1000

Speed	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
Data	0	275	544	813	1082	1351	1620	1889	2158	2427	2696	2965	3234	3503	3772	4041	4310	4579	4848	5117	5386	5655	5924	6193	6462	6731	7000



$Speed_1$=20*(1-0.3*0.6)= 16.4
$Speed_3$ =26*(1-0.3*0.6)= 21.32
$Total\_Data_1$ = 4310
$Total\_Data_3$ = 4200

$Speed_4$ =20*(1-0.2*0)= 20
$Total\_Data_4$ = 5386

$Speed_4$ =20*(1-0.2*0)= 20
$Total\_Data_4$ = 10000

$Speed_1$ =20*(1-0.3*0)= 20
$Total\_Data_1$ = 5000

$Speed_2$ =15*(1-0.25*0)= 15
$Total\_Data_2$ = 3500

$$Avg\_Speed_1 = \frac{(16.4 + 20)}{2} = 18.2$$

$$Avg\_Speed_2 = 15$$

$$Avg\_Speed_3 = 21.32$$

$$Avg\_Speed_4 = \frac{(20 + 20)}{2} = 20$$

$$\text{BestSpeedUsers} = 20 + 15 * 2 + 26 + 20 * 3 = 136$$

$$\text{Objective\_function} = \sum_{\forall\ Users} Avg\_Speed_i\ W_i\ / \text{BestSpeedUsers}$$

= 18.2*1 + 15*2 + 21.32*1 + 20*3 = 129.52/136 = 0.9523

$\sum_{\forall\ Users} Data\_Loss_i$ = 0

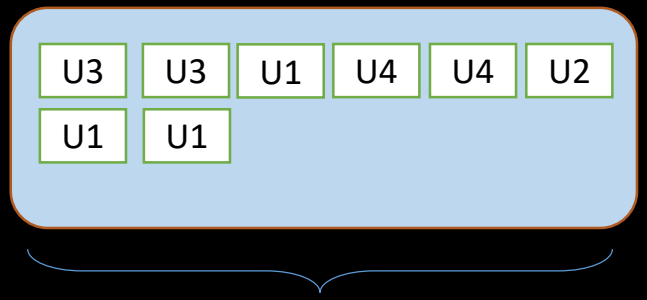
$$\text{Goal\_function} = \text{Objective\_function} - \alpha * \frac{\sum_{\forall\ Users} Data\_Loss_i}{Total\_Data\_of\_All\_Users}$$ = 0.9523 – 1000*0 = 0.9523

Indeed, it is a feasible solution that achives 95.2% of the maximum speed

Example 2

- U1={20,9000,0.3,1}, U2={15,8000,0.25,2}, U3={26,8000,0.6,1}, U4={20,10000,0.2,3}; α = 1000

Speed	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
Data	0	275	544	813	1082	1351	1620	1889	2158	2427	2696	2965	3234	3503	3772	4041	4310	4579	4848	5117	5386	5655	5924	6193	6462	6731	7000



$Speed_1$ =20*(1-0.3*0.6)= 16.4
$Speed_3$ =26*(1-0.3*0.6)= 21.32
$Total\_Data_1$ = 4310
$Total\_Data_3$ = 5655

$Speed_4$ =20*(1-0.2*0)= 20
$Total\_Data_4$ = 5386

$Speed_1$ =20*(1-0.3*0.6)= 16.4
$Speed_3$=26*(1-0.3*0.6)= 21.32
$Total\_Data_1$ = 4310
$Total\_Data_3$ = 8000

$Speed_4$ =20*(1-0.2*0)= 20
$Total\_Data_4$ = 10000

$Speed_1$ =20*(1-0.3*0)= 20
$Total\_Data_1$ = 9000

$Speed_2$ =15*(1-0.25*0)= 15
$Total\_Data_2$ = 4041

$$Avg\_Speed_1 = \frac{(16.4 + 16.4 + 20)}{3} = 17.6$$

$$Avg\_Speed_2 = 15$$

$$Avg\_Speed_3 = \frac{(21.32 + 21.32)}{2} = 21.32$$

$$Avg\_Speed_4 = \frac{(20 + 20)}{2} = 20$$

$$Data\_Loss_{U2} = 8000 - 4041 = 3959$$

$$\sum_{\forall\, Users} Data\_Loss_i = 0 + 3959 + 0 + 0 = 3959$$

$Total\_Data\_of\_All\_Users$ = 9000+8000+8000+10000= 35000

$$Penalty\_term = \frac{\sum_{\forall\, Users} Data\_Loss_i}{Total\_Data\_of\_All\_Users}$$ = (3959/35000) = 0.113

$$\text{Objective\_function} = \sum_{\forall\, Users} Avg\_Speed_i \,/ BestSpeedUsers$$
=(17.6+2*15+21.32+3*20)/136 = 128.92/136 = 0.947

$$\text{Score} = \text{Objective\_function} - \alpha * Penalty\_term$$
= 0.947 – 0.113 * 1000 = -112.053

Indeed, it is an infeasible solution that couldn't send 11.3% of all users' data