TOTAL				BONUSES	Project EVALUATION			100%	BONUSES					DOCUMENTATION				COMPLEXITY				RUN (Complete Cases) (OUTPUT + TIME) Use beyond compare text-comparator to check										RUN (Sample Cases) Use beyond compare text-comparator to check								DELIVERY	ID	G#
BONUS Per Member	MARK (= 15)	TARGET MARK BASED ON #MEMBERS	Percntage (%)	(+)					10%		20%	10%	10%	Should be detailed & correct according to the written code (NOT to the expected)				(E: Roads, V: Intersections, E': Roads discovered, V': Intersections discovered, S: Start nodes, VERT_PATH: Vertices of the shortest path)				NEW LARGE CASE (NAMap.txt - 1000 Queries)			NEW MEDIUM CASE (TGMap.txt - 200 Queries)			OLD LARGE CASE (SFMap.txt - 1000 Queries)		OLD MEDIUM CASE (OLMap.txt - 1000 Queries)		new map8.txt queries8.txt		map5.txt queries5.txt		map4.txt queries4.txt		map1.txt query1.txt				
				% from total mark	DOC	COMPLEXITY	RUN (Complete Cases)	RUN (Sample Cases)	Others (...% out of 100%)		B3: Faster implementation (...% out of 100%)	B2: Visualization (...% out of 100%)	B1: Changing-speed handling (...% out of 100%)	Description of graph construction	Constrcut Path (Backtracking)	Find shortest time path	Graph Construction	Construct Path (Backtracking)	Find shortest time path	Search for possible START & END nodes	Graph Construction		Time & Distances of shortest path	Nodes of shortest path		Time & Distances of shortest path	Nodes of shortest path	Time & Distances of shortest path	Nodes of shortest path	Time & Distances of shortest path	Nodes of shortest path	Time & Distances of shortest path	Nodes of shortest path	Time & Distances of shortest path	Nodes of shortest path	Time & Distances of shortest path	Nodes of shortest path	Time & Distances of shortest path	Nodes of shortest path	SOURCE CODE		
				%	15.0%	15.0%	50.0%	20.0%	Evaluation	Description	super node (1 src/dst): 20% find sources in less than V: 20% Fibonacci heap Fast:40% Bidirectional dijkstra Fast: 40% A* Fast: 40%	25%:draw map, 25%:draw path 25%: query via UI, 25%:zoom in/out	1. Bonus Sample Case (20%: path, 20% time & dists) 2. Bonus Medium Case (30%: path, 30% time & dists)	Exist or Not	1: Detailed & Correct 0: Not Detailed/Correct			O(VERT_PATH)	O(S E` log V')	O(V)	O(E)	TIME (in SECs)	1000 Correct	1000 Correct Time: <= 180 Sec	TIME (in SECs)	200 Correct	200 Correct Time: <= 5 Sec	1000 Correct	1000 Correct Time: <= 180 Sec	1000 Correct	1000 Correct Time: <= 5 Sec	10 Correct	10 Correct Time: < 1 Sec	10 Correct	10 Correct Time: < 1 Sec	10 Correct	10 Correct Time: < 1 Sec	1 Correct	1 Correct Time: < 1 Sec			
0	5.9	15	39.00%	0%	75%	100%	0%	62.5%	0.00%		0.00%	0.00%	0.00%	1	1	1	0	1	1	1	1		0	0		0	0	0	0	0	0	1	1	0	0	0	1	1	1	YES	2021170257 2022170012 2022170238 2022170652 20201700381 20201700738	1
0.7	15	15	128.00%	27.5%	100%	100%	100%	100%	80.00%	Very FAST, the large case runs in 13 sec, new medium 0.2 sec, new large 4 sec, multi-threading using openp for constructing graph, read file fin and fout	60.00%	75.00%	0.00%	1	1	1	1	1	1	1	1	4 sec	1	1	0.2 sec	1	1	1	1	1	1	1	1	1	1	1	1	1	1	YES	2022170006 2022170018 2022170036 2022170264 2022170445 2022170539	3
0.4	15	15	114.00%	14%	100%	100%	100%	100%	20.00%	fast, large case runs in 48 sec	60.00%	0.00%	0.00%	1	1	1	1	1	1	1	1	31 sec	1	1	1.5 sec	1	1	1	1	1	1	1	1	1	1	1	1	1	1	YES	2022170083 2022170414 2022170434 2022170435 2022170466 2022170565	4
0.6	15	15	125.00%	25%	100%	100%	100%	100%	100.00%	construction of this grid takes O (V), implemented Priority queue from scratch and enhanced very much on enqueue and deqeue in O(logN), because the previous enqueue fits to many use cases due to priority changing for nodes	50.00%	50.00%	0.00%	1	1	1	1	1	1	1	1	16 sec	1	1	1.5 sec	1	1	1	1	1	1	1	1	1	1	1	1	1	1	YES	2022170017 2022170145 2022170356 2022170359 2022170365 2022170367	5
0.2	15	15	109.00%	9%	100%	100%	100%	100%	0.00%		20.00%	50.00%	0.00%	1	1	1	1	1	1	1	1	47 sec	1	1	1.7 sec	1	1	1	1	1	1	1	1	1	1	1	1	1	1	YES	2022170063 2022170097 2022170104 2022170213 2022170301 2022170418	8
0.6	15	15	122.00%	22%	100%	100%	100%	100%	40.00%	Very fast , old medium: 0.3 sec. old large: 13 sec	40.00%	100.00%	0.00%	1	1	1	1	1	1	1	1	5 sec	1	1	0.25 sec	1	1	1	1	1	1	1	1	1	1	1	1	1	1	YES	2021170502 2022170175 2022170379 2022170478 2022170479 2022170482	11
0.6	15	15	119.00%	19%	100%	100%	100%	100%	0.00%		50.00%	60.00%	30.00%	1	1	1	1	1	1	1	1	62 sec	1	1	3 sec	1	1	1	1	1	1	1	1	1	1	1	1	1	1	YES	2022170199 2022170240 2022170348 2022170370 2022170584	12
0.1	15	15	104.00%	4%	100%	100%	100%	100%	0.00%		20.00%	0.00%	0.00%	1	1	1	1	1	1	1	1	65 sec	1	1	0.7 sec	1	1	1	1	1	1	1	1	1	1	1	1	1	1	YES	2022170068 2022170095 2022170235 2022170274 2022170415 2022170471	13
0	5.7	15	38.00%	3%	100%	100%	0%	25%	30.00%		0.00%	0.00%	0.00%	1	1	1	1	1	1	1	1	16 sec	0	0	0.5 sec	0	0	0	0	0	0	0	0	0	0	0	0	1	1	YES	2022170038 2022170194 2022170417 2022170461 2022170562 2022170569	16
0.2	15	15	108.00%	8%	100%	100%	100%	100%	0.00%		20.00%	40.00%	0.00%	1	1	1	1	1	1	1	1	150 sec	1	1	3 sec	1	1	1	1	1	1	1	1	1	1	1	1	1	1	YES	2022170053 2022170093 2022170126 2022170135 2022170146 2022170176	17
0.3	15	15	112.00%	11.5%	100%	100%	100%	100%	0.00%	FAST	20.00%	75.00%	0.00%	1	1	1	1	1	1	1	1	22 sec	1	1	0.8 sec	1	1	1	1	1	1	1	1	1	1	1	1	1	1	YES	2022170292 2022170394 2022170396 2022170598 2022170613 2022170673	23
0.1	15	15	104.00%	4%	100%	100%	100%	100%			20.00%	0.00%	0.00%	1	1	1	1	1	1	1	1	49 sec	1	1	2 sec	1	1	1	1	1	1	1	1	1	1	1	1	1	1	YES	2022170107 2022170116 2022170247 2022170488 2022170501 2022170579	25
0.5	15	15	117.00%	17%	100%	100%	100%	100%	30.00%	very fast, tried multi-threading, enters any source and target from the UI and also accept files	20.00%	100.00%	0.00%	1	1	1	1	1	1	1	1	14 sec	1	1	0.5 sec	1	1	1	1	1	1	1	1	1	1	1	1	1	1	YES	2022170011 2022170200 2022170202 2022170237 2022170239	26
0.3	15	15	111.00%	11%	100%	100%	100%	100%	10.00%	made multi-threading in some parts (mainly in GUI)	0.00%	100.00%	0.00%	1	1	1	1	1	1	1	1	28 sec	1	1	1.5 sec	1	1	1	1	1	1	1	1	1	1	1	1	1	1	YES	2022170087 2022170114 2022170317 2022170339 2022170581 2022170625	28
0.5	15	15	112.00%	12%	100%	100%	100%	100%	0.00%		60.00%	0.00%	0.00%	1	1	1	1	1	1	1	1	41 sec	1	1	1.8 sec	1	1	1	1	1	1	1	1	1	1	1	1	1	1	YES	2022170026 2022170278 2022170287 2022170458	88
0.2	15	12	83.00%	17.5%	75%	100%	37.5%	100%	20.00%		40.00%	75.00%	0.00%	1	0	1	1	1	1	1	1	> 5 min	0	0	23 sec	1	0	1	0	1	0	1	1	1	1	1	1	1	1	YES	2022170224 2022170236 2022170280	100
0.3	15	15	110.00%	10%	100%	100%	100%	100%	60.00%	- Parallel code. - Well Documented.	20.00%	0.00%	0.00%	1	1	1	1	1	1	1	1	9.1 sec	1	1	1.2 sec	1	1	1	1	1	1	1	1	1	1	1	1	1	1	YES	2022170030 2022170187 2022170205 2022170233 2022170449 2022170631	47
0.6	15	15	122.00%	22%	100%	100%	100%	100%	50.00%	- Parallel code. - Well Documented.	40.00%	50.00%	40.00%	1	1	1	1	1	1	1	1	24 sec	1	1	0.9 sec	1	1	1	1	1	1	1	1	1	1	1	1	1	1	YES	2022170067 2022170084 2022170129 2022170150 2022170203 2022170214	51
0.8	15	15	125.00%	25%	100%	100%	100%	100%	30.00%	- Well Documented.	60.00%	100.00%	0.00%	1	1	1	1	1	1	1	1	70.3 sec	1	1	2.5 sec	1	1	1	1	1	1	1	1	1	1	1	1	1	1	YES	2022170162 2022170303 2022170340 2022170447 2022170629	55
0.6	15	15	119.00%	19%	100%	100%	100%	100%	20.00%	- Well Documented.	40.00%	50.00%	40.00%	1	1	1	1	1	1	1	1	52.5 sec	1	1	1.4 sec	1	1	1	1	1	1	1	1	1	1	1	1	1	1	YES	2022170113 2022170206 2022170441 2022170472 2022170611	57
0.7	15	15	126.00%	26%	100%	100%	100%	100%	80.00%	- Dynamic rendering for multiple queries in UI. - Selecting any map and/or query files. - Well Documented. - Interactive UI.	40.00%	100.00%	0.00%	1	1	1	1	1	1	1	1	21.2 sec	1	1	1.1 sec	1	1	1	1	1	1	1	1	1	1	1	1	1	1	YES	2022170051 2022170071 2022170075 2022170090 2022170305 2022170307	60
0.8	15	15	131.00%	31%	100%	100%	100%	100%	50.00%	- Well Documented. - Interactive UI.	80.00%	100.00%	0.00%	1	1	1	1	1	1	1	1	11 sec	1	1	0.6 sec	1	1	1	1	1	1	1	1	1	1	1	1	1	1	YES	2022170055 2022170077 2022170204 2022170209 2022170225 2022170483	62
0.2	15	15	108.00%	8%	100%	100%	100%	100%	40.00%	- Pirority Queue.	20.00%	0.00%	0.00%	1	1	1	1	1	1	1	1	136.9 sec	1	1	2.7 sec	1	1	1	1	1	1	1	1	1	1	1	1	1	1	YES	2022170040 2022170098 2022170369 2022170381 2022170421 20201701024	64
0.2	15	15	106.00%	6%	100%	100%	100%	100%	20.00%	- K-Dimenional tree in Graph Construction.	20.00%	0.00%	0.00%	1	1	1	1	1	1	1	1	40 sec	1	1	2 sec	1	1	1	1	1	1	1	1	1	1	1	1	1	1	YES	2022170021 2022170042 2022170172 2022170261 2022170266 2022170286	65
0.3	15	15	111.00%	11%	100%	100%	100%	100%	30.00%	- Very Fast for new Large.	40.00%	0.00%	0.00%	1	1	1	1	1	1	1	1	7.3 sec	1	1	0.32 sec	1	1	1	1	1	1	1	1	1	1	1	1	1	1	YES	2022170315 2022170320 2022170325 2022170329 2022170515 2022170518	66
0.7	15	15	123.00%	23%	100%	100%	100%	100%	50.00%	- Well Documented. - Find shortest path using mouse via UI. - Multi-source bi-directional Dijkstra.	40.00%	100.00%	0.00%	1	1	1	1	1	1	1	1	35.9 sec	1	1	1.7 sec	1	1	1	1	1	1	1	1	1	1	1	1	1	1	YES	2022170324 2022170457 2023170341 2023170749 20201700430	67
										- Draw All shortest paths for all																															2022170041	

TOTAL				BONUSES	Project EVALUATION			100%	BONUSES					DOCUMENTATION				COMPLEXITY				RUN (Complete Cases) (OUTPUT + TIME) Use beyond compare text-comparator to check										RUN (Sample Cases) Use beyond compare text-comparator to check								DELIVERY	ID	G#
BONUS Per Member	MARK (= 15)	TARGET MARK BASED ON #MEMBERS	Percntage (%)	(+)					10%		20%	10%	10%	Should be detailed & correct according to the written code (NOT to the expected)				(E: Roads, V: Intersections, E': Roads discovered, V': Intersections discovered, S: Start nodes, VERT_PATH: Vertices of the shortest path)				NEW LARGE CASE (NAMap.txt - 1000 Queries)			NEW MEDIUM CASE (TGMap.txt - 200 Queries)			OLD LARGE CASE (SFMap.txt - 1000 Queries)		OLD MEDIUM CASE (OLMap.txt - 1000 Queries)		new map8.txt queries8.txt		map5.txt queries5.txt		map4.txt queries4.txt		map1.txt query1.txt				
				% from total mark	DOC	COMPLEXITY	RUN (Complete Cases)	RUN (Sample Cases)	Others (...% out of 100%)		B3: Faster implementation (...% out of 100%)	B2: Visualization (...% out of 100%)	B1: Changing-speed handling (...% out of 100%)	Description of graph construction	Constrcut Path (Backtracking)	Find shortest time path	Graph Construction	Construct Path (Backtracking)	Find shortest time path	Search for possible START & END nodes	Graph Construction		Time & Distances of shortest path	Nodes of shortest path		Time & Distances of shortest path	Nodes of shortest path	Time & Distances of shortest path	Nodes of shortest path	Time & Distances of shortest path	Nodes of shortest path	Time & Distances of shortest path	Nodes of shortest path	Time & Distances of shortest path	Nodes of shortest path	Time & Distances of shortest path	Nodes of shortest path	Time & Distances of shortest path	Nodes of shortest path	SOURCE CODE		
				%	15.0%	15.0%	50.0%	20.0%	Evaluation	Description	super node (1 src/dst): 20% find sources in less than V: 20% Fibonacci heap Fast:40% Bidirectional dijkstra Fast: 40% A* Fast: 40%	25%:draw map, 25%:draw path 25%: query via UI, 25%:zoom in/out	1. Bonus Sample Case (20%: path, 20% time & dists) 2. Bonus Medium Case (30%: path, 30% time & dists)	Exist or Not	1: Detailed & Correct 0: Not Detailed/Correct			O(VERT_PATH)	O(S E` log V')	O(V)	O(E)	TIME (in SECs)	1000 Correct	1000 Correct Time: <= 180 Sec	TIME (in SECs)	200 Correct	200 Correct Time: <= 5 Sec	1000 Correct	1000 Correct Time: <= 180 Sec	1000 Correct	1000 Correct Time: <= 5 Sec	10 Correct	10 Correct Time: < 1 Sec	10 Correct	10 Correct Time: < 1 Sec	10 Correct	10 Correct Time: < 1 Sec	1 Correct	1 Correct Time: < 1 Sec			
0.5	15	15	118.00%	18%	100%	100%	100%	100%	50.00%	- Draw All shortest paths for all queries at once. - Enhanced Dijkstra Implementation by applying early stop (i.e., Find shortest paths for end candidates)..	40.00%	50.00%	0.00%	1	1	1	1	1	1	1	1	20.4 sec	1	1	1 sec	1	1	1	1	1	1	1	1	1	1	1	1	1	1	YES	2022170046 2022170062 2022170196 2022170198 2022170290	72
0.6	15	15	123.00%	23%	100%	100%	100%	100%	10.00%		60.00%	100.00%	0.00%	1	1	1	1	1	1	1	1	24.6 sec	1	1	1.1 sec	1	1	1	1	1	1	1	1	1	1	1	1	1	1	YES	2022170025 2022170166 2022170215 2022170388 2022170489 2022170522	74
0.5	15	15	121.00%	21%	100%	100%	100%	100%	30.00%	- Well Documented.	40.00%	100.00%	0.00%	1	1	1	1	1	1	1	1	10.2 sec	1	1	1.7 sec	1	1	1	1	1	1	1	1	1	1	1	1	1	1	YES	2021170511 2022170102 2022170128 2022170309 2022170510 2022170536	77
0.8	15	15	131.00%	31%	100%	100%	100%	100%	100.00%	- Well Documented. - Draw All shortest paths for all queries at once. - Multithreaded for queries. - Applied Game Collision Algorithm to reduce calculations of start and end nodes.	60.00%	50.00%	40.00%	1	1	1	1	1	1	1	1	2.6 sec	1	1	0.17 sec	1	1	1	1	1	1	1	1	1	1	1	1	1	1	YES	2022170353 2022170373 2022170375 2022170385 2022170389 2022170456	83
0.6	15	15	121.00%	20.5%	100%	100%	100%	100%	50.00%	- Parallel code. - K-Dimenional tree in Nearest neighbors searching resulting in lg|V| time complexity.	40.00%	75.00%	0.00%	1	1	1	1	1	1	1	1	4.8 sec	1	1	0.38 sec	1	1	1	1	1	1	1	1	1	1	1	1	1	1	YES	2021170692 2022170485 2022170493 2022170630 2022170748	85
0.6	15	15	122.00%	21.5%	100%	100%	100%	100%	20.00%	- Very Fast New Large.	60.00%	10.00%	65.00%	1	1	1	1	1	1	1	1	7.7 sec	1	1	0.6 sec	1	1	1	1	1	1	1	1	1	1	1	1	1	1	YES	2022170241 2022170308 2022170436 2022170484 2022170491 2022170634	87
0.4	15	15	114.00%	14%	100%	100%	100%	100%	40.00%	Very very fast andmade parallel processing upon queries	20.00%	60.00%	0.00%	1	1	1	1	1	1	1	1	5 sec	1	1	0.7 sec	1	1	1	1	1	1	1	1	1	1	1	1	1	1	YES	2022170327 2022170328 2022170372 2022170596 2022170609 2022170624	29
0.3	15	15	112.00%	12%	100%	100%	100%	100%	0.00%		20.00%	80.00%	0.00%	1	1	1	1	1	1	1	1	50 sec	1	1	0.6 sec	1	1	1	1	1	1	1	1	1	1	1	1	1	1	YES	2021170521 2022170420 2022170430 2022170498 2022170601 20201700038	30
0.1	15	15	104.00%	8%	75%	100%	100%	100%	0.00%		40.00%	0.00%	0.00%	1	0	1	1	1	1	1	1	35 sec	1	1	1 sec	1	1	1	1	1	1	1	1	1	1	1	1	1	1	YES	2022170035 2022170037 2022170127 2022170134 2022170139 2022170211	35
0.2	15	15	109.00%	9%	100%	100%	100%	100%	0.00%		20.00%	50.00%	0.00%	1	1	1	1	1	1	1	1	1.6 min	1	1	2.3 sec	1	1	1	1	1	1	1	1	1	1	1	1	1	1	YES	2022170057 2022170568 2022170591 2022170626 2022170633 2022170654	36
0.4	15	15	116.00%	16%	100%	100%	100%	100%	30.00%	Very fast , optimized some parts as calculating distance	20.00%	50.00%	40.00%	1	1	1	1	1	1	1	1	5.5 sec	1	1	0.3 sec	1	1	1	1	1	1	1	1	1	1	1	1	1	1	YES	2022170183 2022170360 2022170378 2022170387 2022170397 2022170527	37
0.1	15	15	104.00%	4%	100%	100%	100%	100%	0.00%		20.00%	0.00%	0.00%	1	1	1	1	1	1	1	1	20 sec	1	1	1 sec	1	1	1	1	1	1	1	1	1	1	1	1	1	1	YES	2022170350 2022170399 2022170404 2022170496 20201700229 20201700317	39
0.5	15	15	115.00%	15%	100%	100%	100%	100%	40.00%	made custom indexed priotrity queue by overriding some methods to be more effiecent, by replacing the vertex distance instead of removing and adding each time	20.00%	70.00%	0.00%	1	1	1	1	1	1	1	1	43 sec	1	1	1 sec	1	1	1	1	1	1	1	1	1	1	1	1	1	1	YES	2022170094 2022170105 2022170152 2022170487 2022170494	40
0.1	15	15	103.00%	15%	100%	100%	75%	100%	10.00%		20.00%	100.00%	0.00%	1	1	1	1	1	1	1	1	3 min	1	1	4.6 sec	1	1	1	0	1	0	1	1	1	1	1	1	1	1	YES	2022170160 2022170192 2022170433 2022170437 2022170468 2022170469	41
0	14.1	15	94.00%	6%	100%	100%	75%	100%	20.00%	Fast time	20.00%	0.00%	0.00%	1	1	1	1	1	1	1	1	15 sec	1	1	0.3 sec	1	1	1	1	0	0	1	1	1	1	1	1	1	1	YES	2022170020 2022170023 2022170123 2022170559 2022170583 2022170585	42
0.3	15	15	113.00%	13%	100%	100%	100%	100%	40.00%	very detailed report and best till now. analyzing every line regarding space and time, multi-threading on queries	20.00%	50.00%	0.00%	1	1	1	1	1	1	1	1	23 sec	1	1	1.2 sec	1	1	1	1	1	1	1	1	1	1	1	1	1	1	YES	2022170227 2022170228 2022170351 2022170665 2023170238 2023170347	43
0.6	15	15	123.00%	23%	100%	100%	100%	100%	10.00%	Fast, optimized dijkstra somehow, good analysis	60.00%	60.00%	40.00%	1	1	1	1	1	1	1	1	23 sec	1	1	0.5 sec	1	1	1	1	1	1	1	1	1	1	1	1	1	1	YES	2022170064 2022170066 2022170089 2022170155 2022170245 2022170407	45
0.9	15	15	135.00%	35%	100%	100%	100%	100%	90.00%	multi-threading over queries, fastest time till now, super node , A*, best GUI till now handling any ozoom like google maps and show detasils on the highest level of zoom like distance , x and y and so on	60.00%	100.00%	40.00%	1	1	1	1	1	1	1	1	3.6 sec	1	1	0.2 sec	1	1	1	1	1	1	1	1	1	1	1	1	1	1	YES	2022170031 2022170033 2022170255 2022170272 2022170276 2022170500	59
0	0	15	0.00%	0%	0%	0%	0%	0%	0.00%		0.00%	0.00%	0.00%	PLZ_DELIVER_1ST!	PLZ_DELIVER_1ST!	PLZ_DELIVER_1ST!	PLZ_DELIVER_1ST!	PLZ_DELIVER_1ST!	PLZ_DELIVER_1ST!	PLZ_DELIVER_1ST!	PLZ_DELIVER_1ST!		PLZ_DELIVER_1ST!	PLZ_DELIVER_1ST!		PLZ_DELIVER_1ST!	PLZ_DELIVER_1ST!	PLZ_DELIVER_1ST!	PLZ_DELIVER_1ST!	PLZ_DELIVER_1ST!	PLZ_DELIVER_1ST!	PLZ_DELIVER_1ST!	PLZ_DELIVER_1ST!	PLZ_DELIVER_1ST!	PLZ_DELIVER_1ST!	PLZ_DELIVER_1ST!	PLZ_DELIVER_1ST!	PLZ_DELIVER_1ST!	PLZ_DELIVER_1ST!	NO	2021170011 2022170190 2022170248 2022170250 20201701203 20201701224	90