						Kulunut aika ms									
Algoritmi	Heap / Dheap(Ikm)	Svötteen koke	Syötteen rakenne	Kulunut aika ms (avg)	Solmu Ikm	1	2	3	4	5	6	7	8	9	10
Dijkstra	2-Heap	10*10	etäisyys vakio, 1	1.8			2	2	2	2	3	2	2	2	1
	211000	100*100	otalogyo valuo, i	0			-		_	_		-	_		
		1000*1000		0											
		2000*2000		0											
		3000*3000		0											
		10*10	etäisyys välillä 1-9	2.8	96,48,139,60,86,65,63,1	4	2	2	3	3	2	3	4	2	3
		100*100	,,		19693,36,644,19757,196	63	2	8	54	62	8	52	4	9	10
		1000*1000			48, 7963,41,7531,19941	24	48	22	48	655	52	593	74	51	668
		2000*2000			50,7995796, 44,7995949		2561	28	2482	223	2740	152	144		
		3000*3000		0											
		10*10	etäisyys välillä 1-99	0											
		100*100		0											
		1000*1000		0											
		2000*2000		0											
		3000*3000		0											
	3-Неар	10*10	etäisyys vakio, 1	0											
	·	100*100	,,,,,,		19566,38,18723,53,1969	47	2	52	2	48	51	10	3	47	1
		1000*1000			34, 5901,41, 5796, 1997	13	40	11	42	712	41	33	38	11	44
		2000*2000			11722, 32204,7994182,7		154	4111	2700	140	159	131	2605	28	29
		3000*3000		0											
		10*10	etäisyys välillä 1-9	0											
		100*100	,,,	0											
		1000*1000		0											
		2000*2000		0											
		3000*3000		0											
		10*10	etäisyys välillä 1-99	0											
		100*100		0											
		1000*1000		0											
		2000*2000		0											
		3000*3000		0											
	PriorityQueue (Java)	10*10	etäisyys vakio, 1	0											
	(.2.2.)	100*100	,	0											
		1000*1000		0											
		2000*2000		0											
		3000*3000		0											
		10*10	etäisyys välillä 1-9	0											
		100*100		0											
		1000*1000		0											
		2000*2000		0											
		3000*3000		0											
		10*10	etäisyys välillä 1-99	0											
		100*100		0	-										
		1000*1000		0											
		2000*2000		0											
		3000*3000		0											
		2200 0000		0											
A*	2-Heap	10*10	etäisyys vakio, 1	0.3		0	0	0	0	1	0	1	0	1	0
	loup	100*100	otalogyo valtio, 1	0.5		0	J		<u> </u>						
		1000*1000		0											
		2000*2000		0											
		3000*3000		0											
		10*10	etäisyys välillä 1-9		33,127,12,20,25,145,65,	1	1	0	0	0	1	2	0	1	2
		100*100			16,22,15,17,23,49,16,13.		0	1	1	1	1	0	1	1	1

						Kulunut aika ms									
Algoritmi	Heap / Dheap(Ikm)	Svötteen koko	Syötteen rakenne	Kulunut aika ms (avg)	Solmu Ikm	ruiunut aika ms	2	3	4	5	6	7	8	9	10
7go	ricup / Bricup(ikiri)	1000*1000	Cyclicen rakenne	24.3		42	27	27	34	7	26	6	42	28	4
		2000*2000			31, 7995905,7827307,18		2697	3250	15	3147	23	20	4475	67	67
		3000*3000		0		21	2031	3230	13	3147	23	20	4473	01	01
		10*10	etäisyys välillä 1-99	0											
		100*100	etaisyys vaillia 1-99	0											
		1000*1000		0											
		2000*2000		0											
		3000*3000		0											
		3000 3000		0											
	2 Hoon	10*10	otäisvus vakio 1	0											
	3-Heap	100*100	etäisyys vakio, 1		59,17,39,15,10,10,59,20,	16	0	1	1	1	1	1	0	50	1
		1000*1000			12, 1997960,1997964,19		565	486	753	25	23	31	4	7	4
		2000*2000			7995973, 7995888,126,5		3974	76	1917	34	14	21	2643	,	
		3000*3000		1149.0		2017	3974	70	1917	34	14	21	2043		
		10*10	otäiovaro välillä 1 0	0											
			etäisyys välillä 1-9	0											
		100*100 1000*1000		0											
		2000*1000		0											
		3000*3000		0											
		10*10	otäiovaro välillä 1 00	0											
		100*100	etäisyys välillä 1-99	0											
		1000*1000		0											
		2000*2000		0											
		3000*3000		0											
	Daisait Occaso (Java)	10*10		0											
	PriorityQueue (Java)	10*10 100*100	etäisyys vakio, 1	0											
		100 100		0											
				0											
		2000*2000		0											
		3000*3000 10*10	-4*:												
		100*100	etäisyys välillä 1-9	0											
		1000*1000		0											
		2000*2000		0											
		3000*3000	-4#1												
		10*10	etäisyys välillä 1-99	0											
		100*100		0											
		1000*1000		0											
		2000*2000													
		3000*3000		0											
IDA*	2 Hoan	10*10	otäisvars vakis 1	-											
IDA*	2-Heap	10*10	etäisyys vakio, 1	0											
		100*100		0											
		1000*1000		-											
		2000*2000		0											
		3000*3000													
		10*10	etäisyys välillä 1-9	0											
		100*100		0											
		1000*1000		0											
		2000*2000		0											
		3000*3000		0											
		10*10	etäisyys välillä 1-99	0											
		100*100		0											

						Kulunut aika ms									
Algoritmi	Heap / Dheap(lkm)	Svötteen koko	Syötteen rakenne	Kulunut aika ms (avg)	Solmu Ikm	Kulunut aika ms	2	3	4	5	6	7	8	9	1
-igorium	rieap / Drieap(ikiri)	1000*1000	Syotteen rakenne	(Calanat aika ilis (avg)					7	•	0		-	3	
		2000*2000			)										
		3000*3000			)										
		3000 3000			)										
	3-Неар	10*10	etäisyys vakio, 1		)										
	Оттеар	100*100	ctaloyyo valdo, 1		)										
		1000*1000													
		2000*2000													
		3000*3000													
		10*10	etäisyys välillä 1-9												
		100*100	ctaloyyo vallia 1 o												
		1000*1000													
		2000*2000													
		3000*3000			)										
		10*10	etäisyys välillä 1-99		)										
		100*100	otaloj je valila 1 ee		)										
		1000*1000			)										
		2000*2000			)										
		3000*3000		(											
				(											
	PriorityQueue (Java)	10*10	etäisyys vakio, 1	(											
		100*100		(											
		1000*1000		(											
		2000*2000		(											
		3000*3000		(											
		10*10	etäisyys välillä 1-9	(											
		100*100		(											
		1000*1000		(											
		2000*2000		(											
		3000*3000		(											
		10*10	etäisyys välillä 1-99	(	)										
		100*100		(	)										
		1000*1000		(											
		2000*2000		(	)										
		3000*3000		(	)										