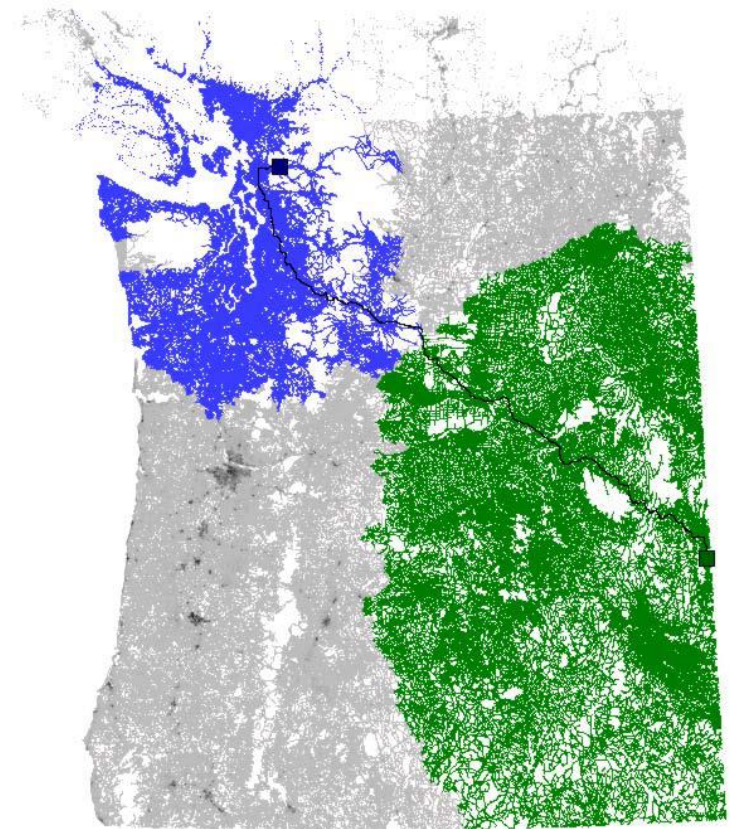
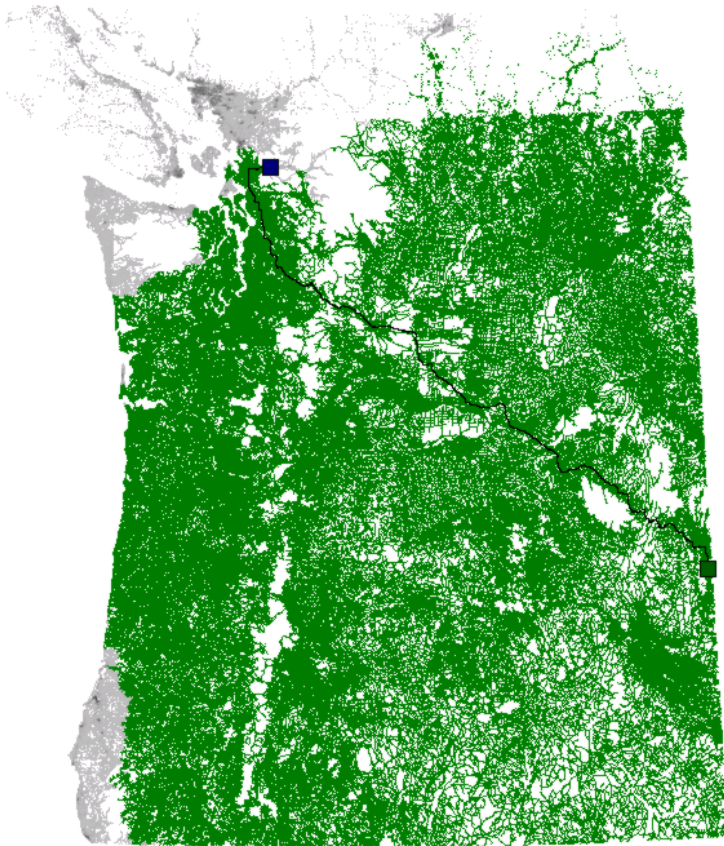


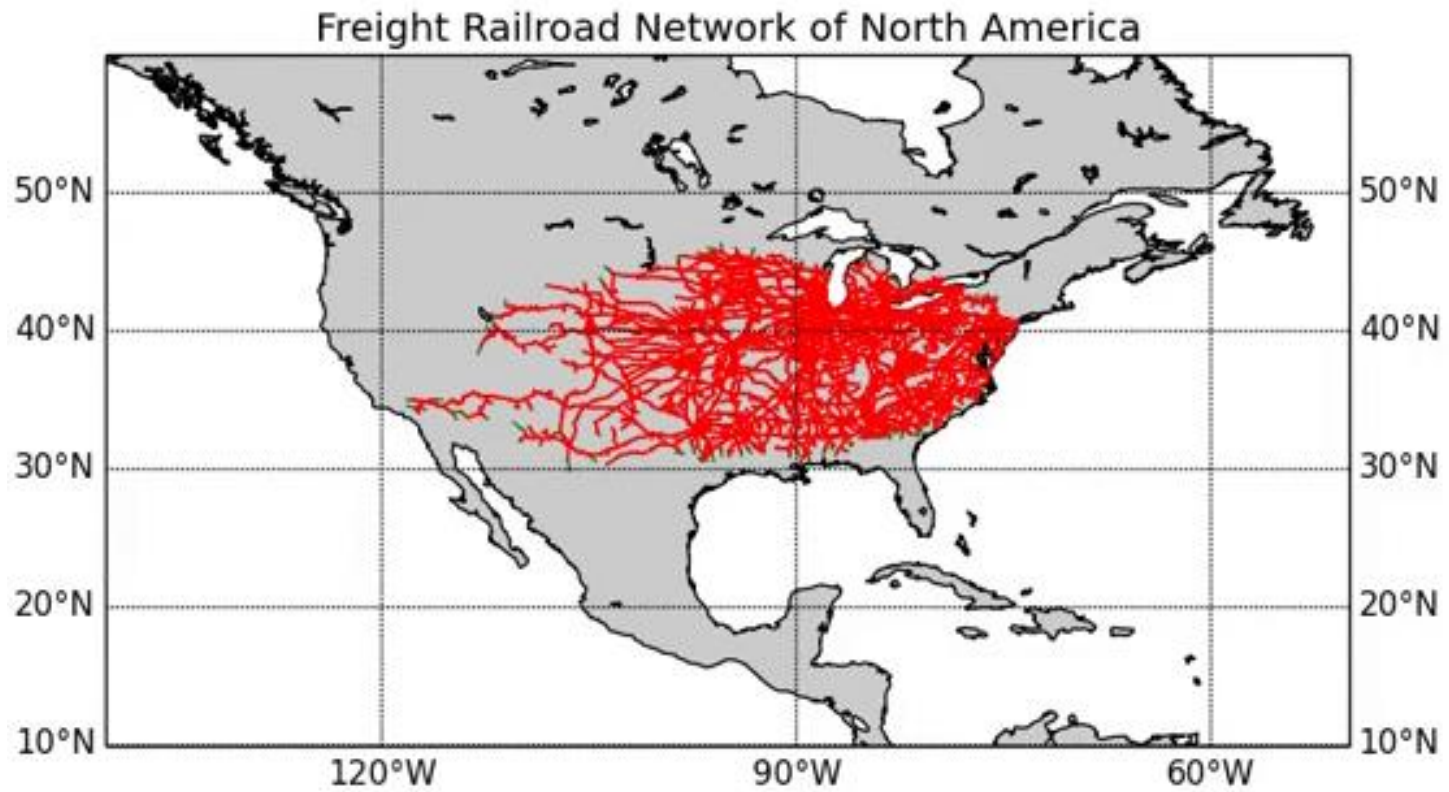
Route Planning

- Tabulation
- Dijkstra
- Bidirectional
- A*
- Landmarks
- Reach
- ArcFlags
- Transit Nodes
- Contraction Hierarchies
- Hub-based labelling

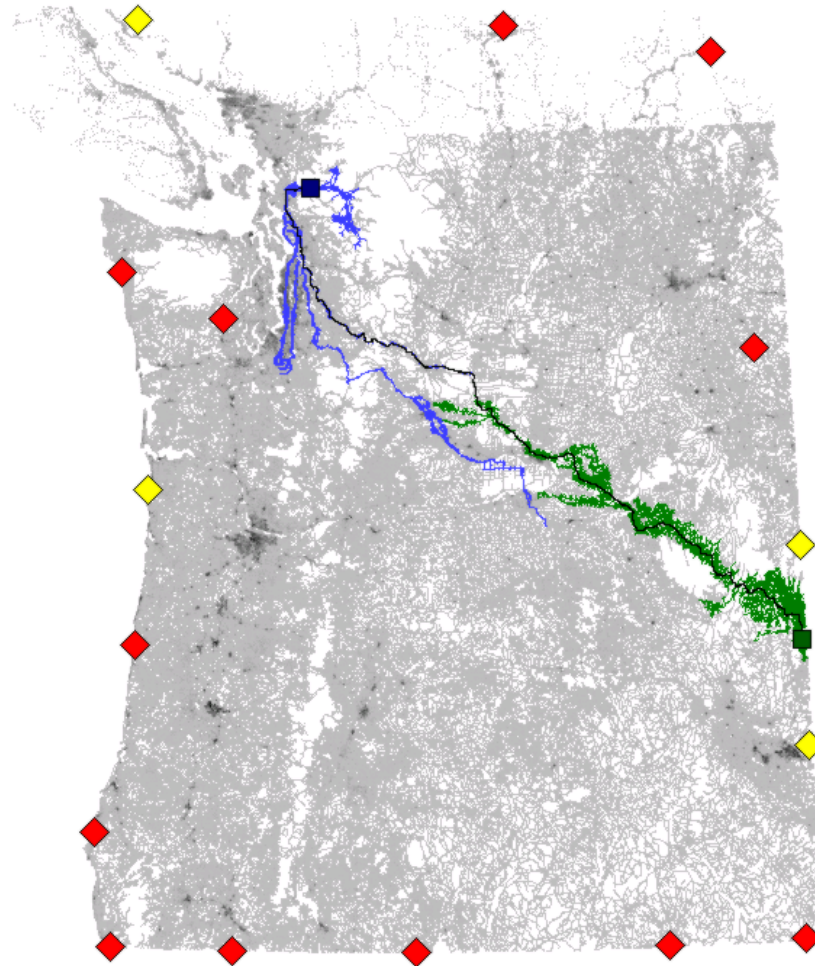
Dijkstra vs Bidirectional Dijkstra



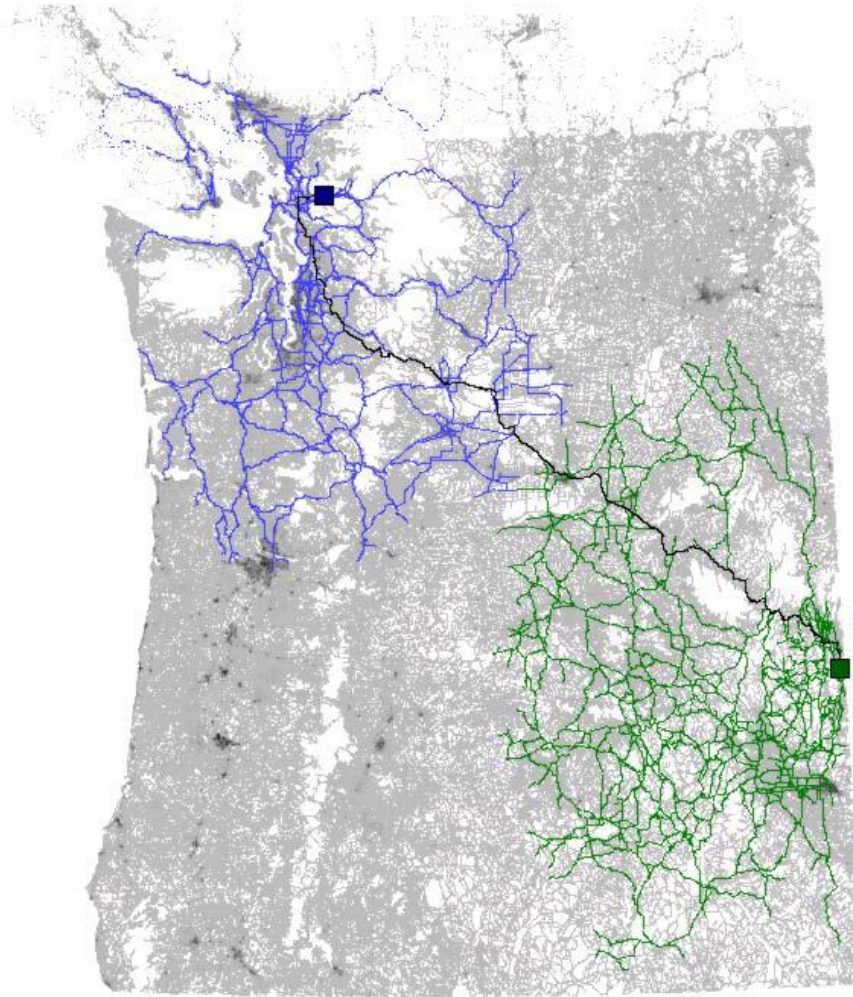
A*



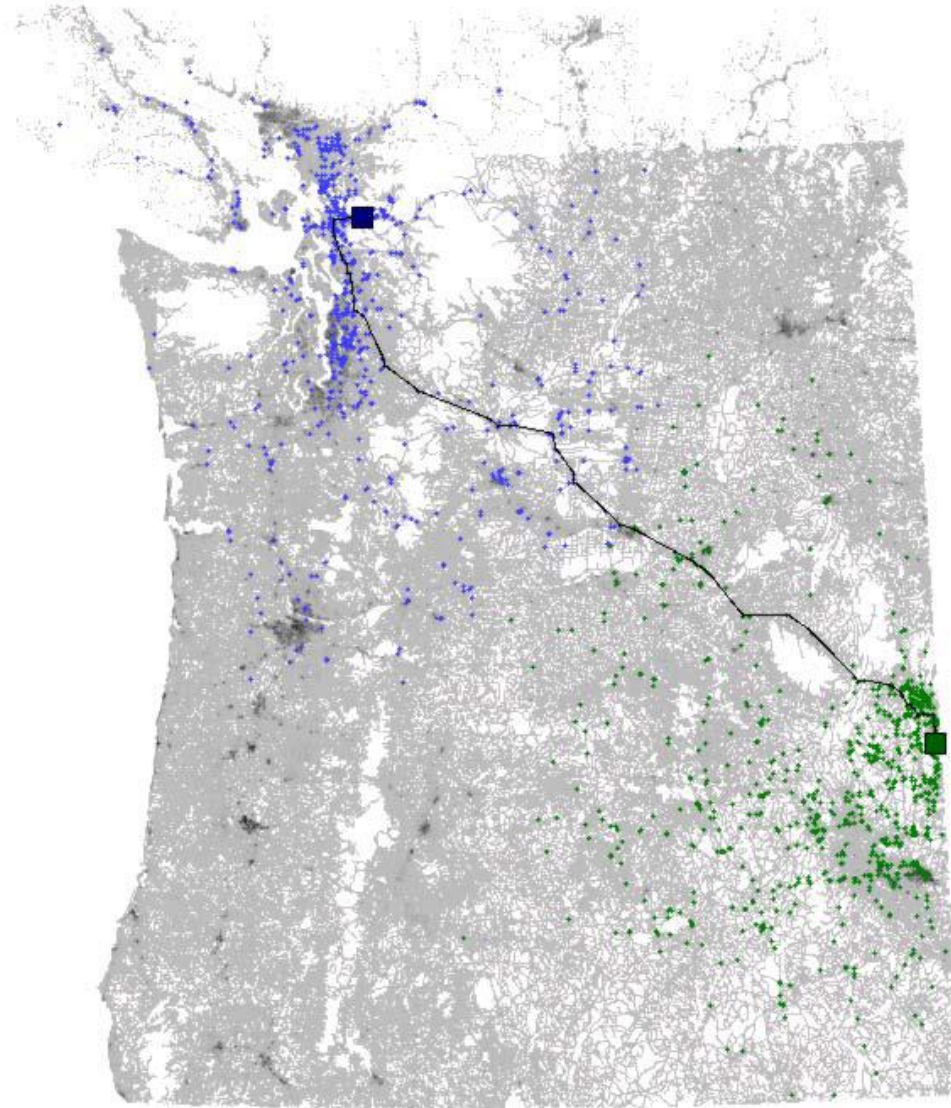
Bidirectional A* with Landmarks



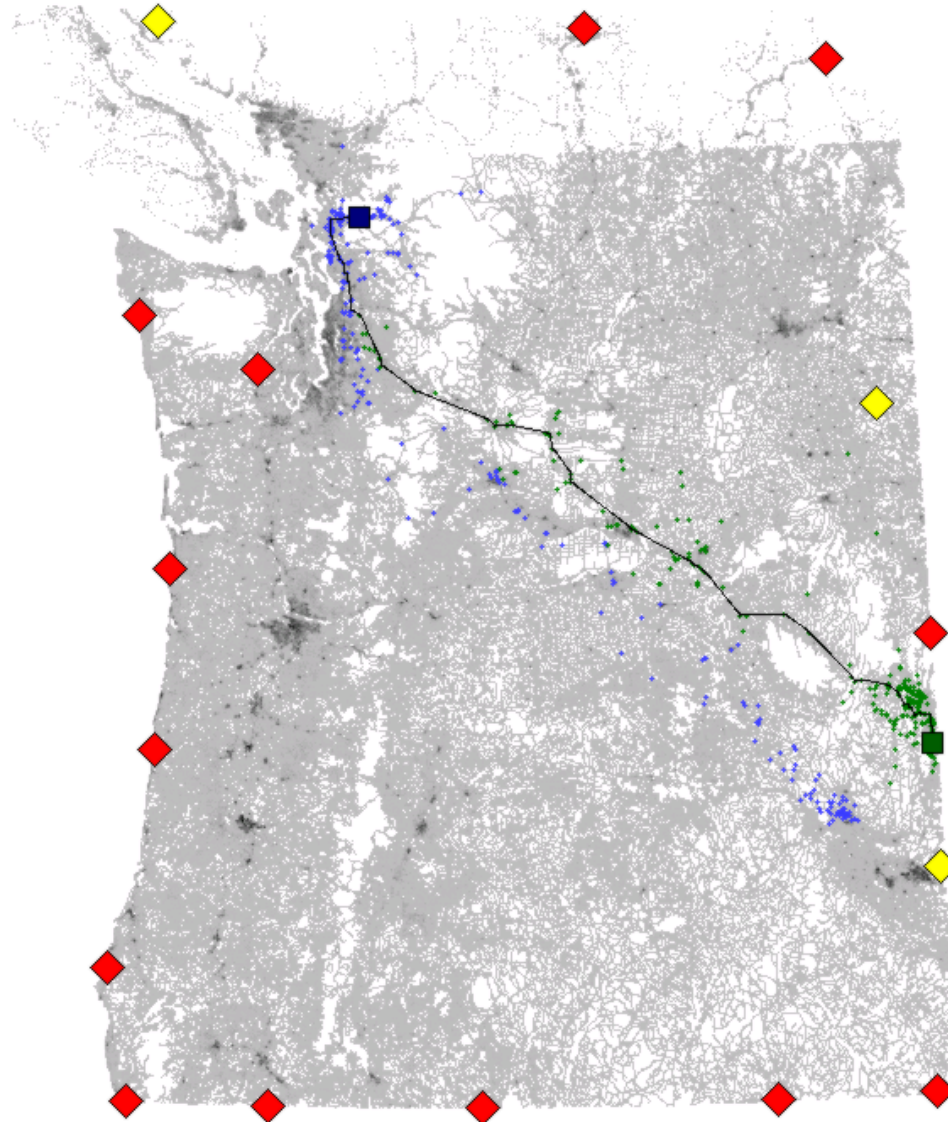
Reach(v)



Reach(v) + Shortcuts



Reach(v) + Shortcuts + Landmarks



Experiments – Northwest US

METHOD	PREPROCESSING		QUERY		
	minutes	MB	avgscan	maxscan	ms
Bidirectional Dijkstra	—	28	518 723	1 197 607	340.74
Landmarks	4	132	16 276	150 389	12.05
Reaches	1100	34	53 888	106 288	30.61
Reaches+Shortcuts	17	100	2 804	5 877	2.39
Reaches+Shortcuts+Landmarks	21	204	367	1 513	0.73

Transit Nodes

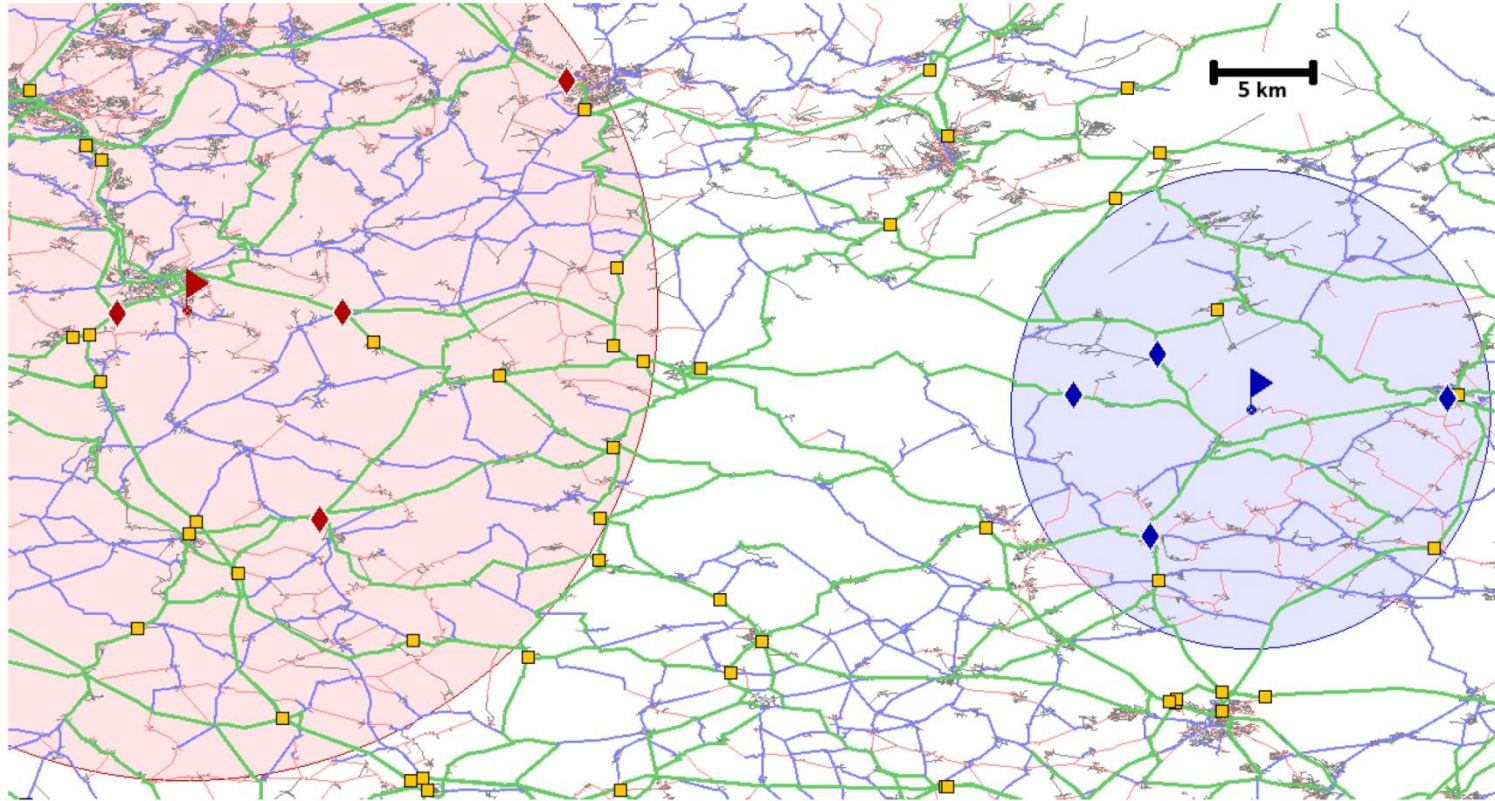


Figure 1: Finding the optimal travel time between two points (flags) somewhere between Saarbrücken and Karlsruhe amounts to retrieving the 2×4 *access nodes* (diamonds), performing 16 table lookups between all pairs of access nodes, and checking that the two disks defining the *locality filter* do not overlap. *Transit nodes* that are not relevant for the depicted query are drawn as small squares.

method	EUROPE			USA			
	preprocessing time [h:m]	space [GB]	query [ns]	preprocessing time [h:m]	space [GB]	query [ns]	
CH [5]	0:13	0.4	93 995	0:14	0.5	67 885	
CHASE [5]	0:52	0.6	9 034	1:59	0.7	9 922	
HPML [9]	≈12:00	3.0	9 817	≈12:00	5.1	10 078	
TNR [5]	0:58	3.7	1 775	0:47	5.4	1 566	
TNR+AF [5]	2:00	5.7	992	1:22	6.3	888	
HL prefix	2:31 + 0:45	5.7	527	2:17 + 0:40	6.4	542	
HL local	2:31 + 0:08	20.1	572	2:17 + 0:07	22.7	627	
HL global	2:31 + 0:14	21.3	276	2:17 + 0:18	25.4	266	
Table Lookup	> 11:03	1 208	358.7	56	> 22:44	2 293 902.1	56