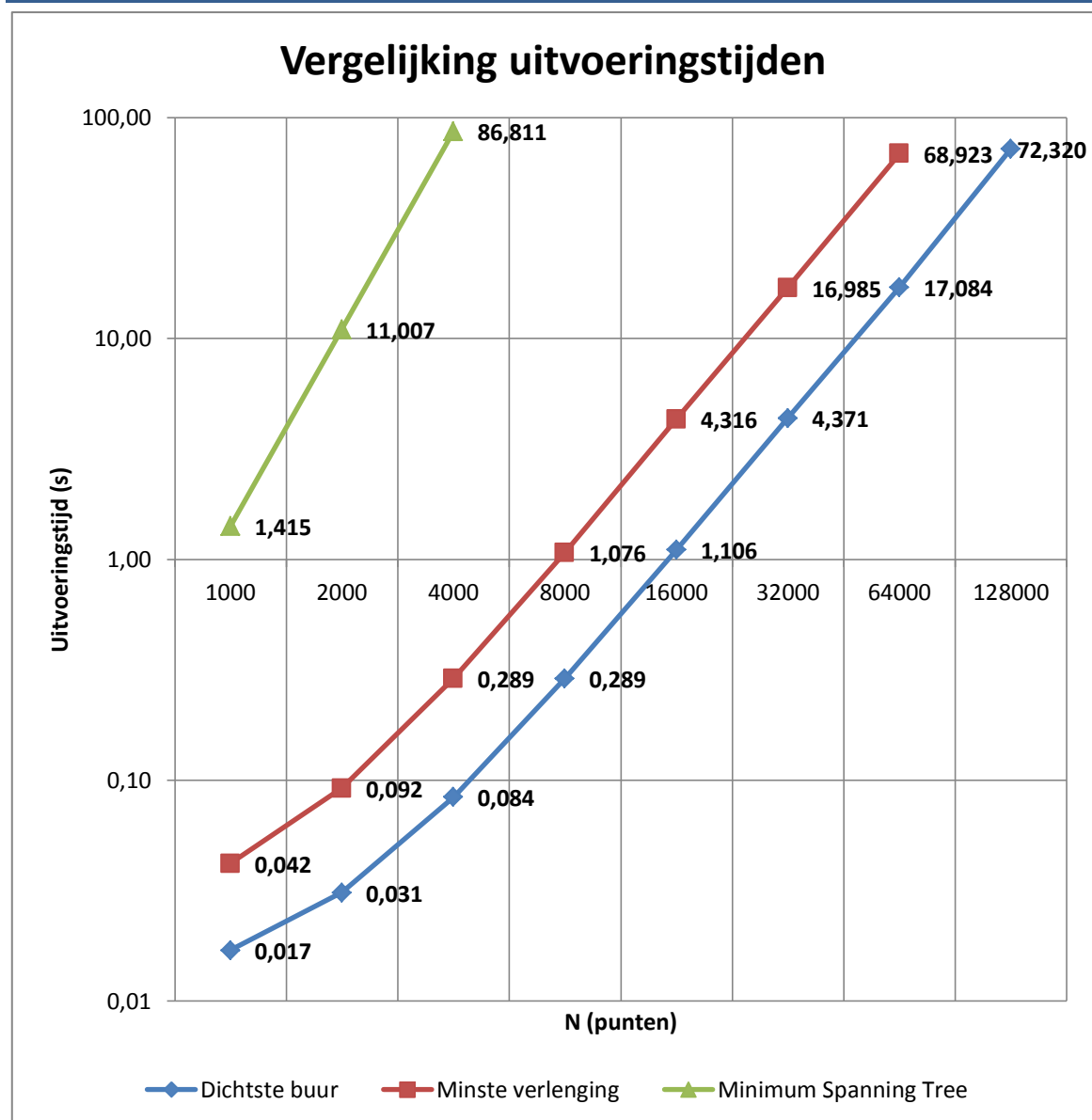


Practicum 4: verslag

Inhoud

1. Grafiek.....	1
2. Tabel.....	2
3. Bespreking.....	2

1. Grafiek



2. Tabel

N	Dichtste buur		Minste verlenging		Minimum Spanning Tree	
	NN: afstand	NN: tijd (s)	SI: afstand	SI: tijd (s)	MST: afstand	MST: tijd (s)
1000	27295,45	0,017	15754,35	0,042	19624,81	1,415
2000	37296,68	0,031	21956,01	0,092	26742,32	11,007
4000	52241,06	0,084	31148,50	0,289	37889,14	86,811
8000	74914,72	0,289	44070,01	1,076	53561,68	767,232
16000	105756,53	1,106	62200,79	4,316	n.v.t.	n.v.t.
32000	148345,21	4,371	87582,57	16,985	n.v.t.	n.v.t.
64000	209286,29	17,084	123983,67	68,923	n.v.t.	n.v.t.
128000	296095,39	72,320	175352,09	276,022	n.v.t.	n.v.t.
256000	420127,84	367,587	n.v.t.	n.v.t.	n.v.t.	n.v.t.

3. Bespreking

We zitten hier met een tweezijdig probleem. Zowel de gevonden totale afstand als de uitvoeringstijd van het algoritme vormt een belangrijke rol bij de beoordeling van de algoritmes. Nu gaan we echter enkel dieper in op de uitvoeringstijden.

Uit de grafiek blijkt duidelijk dat Minimum Spanning Tree (MST), opgebouwd met het algoritme van Prim, het traagste is. Daarna volgt Minste verlenging (SI) en Dichtste buur (NN). SI levert wel telkens de beste tour af.

De tijdscomplexiteit voor NN en SI is telkens $O(N^2)$. Dit heeft te maken met het aantal uitgevoerde for-loops in beide algoritmes. Doordat de constanten verschillen is SI wel sneller.

Bij Prim ligt het iets ingewikkelder. Naargelang het gekozen datatype varieert de uitvoeringstijd per deel van het algoritme. Er moet namelijk gezocht worden naar edges met de kortste afstand, maar anderzijds worden er ook vaak punten toegevoegd en verwijderd uit heaps. Aangezien je qua uitvoeringstijd inboet bij het ene wat je wint bij het andere komt het totaal ook neer op $O(N^2)$. In mijn implementatie heb ik ervoor gekozen om vooral het zoeken snel te houden. Doordat de constanten verschillen is het opbouwen van een MST veel trager dan NN en SI.

Na het opstellen van de MST moet deze ook nog eens in pre-order doorlopen worden. Deze Depth First Search (DFS) heeft een tijdscomplexiteit van $O(N + E)$. Hierbij is N opnieuw gelijk aan het aantal punten en E gelijk aan het aantal edges. In worst case is $E = N^2$.

We kunnen besluiten dat een MST hier geen ideale oplossing is en we beter voor NN zouden kiezen.