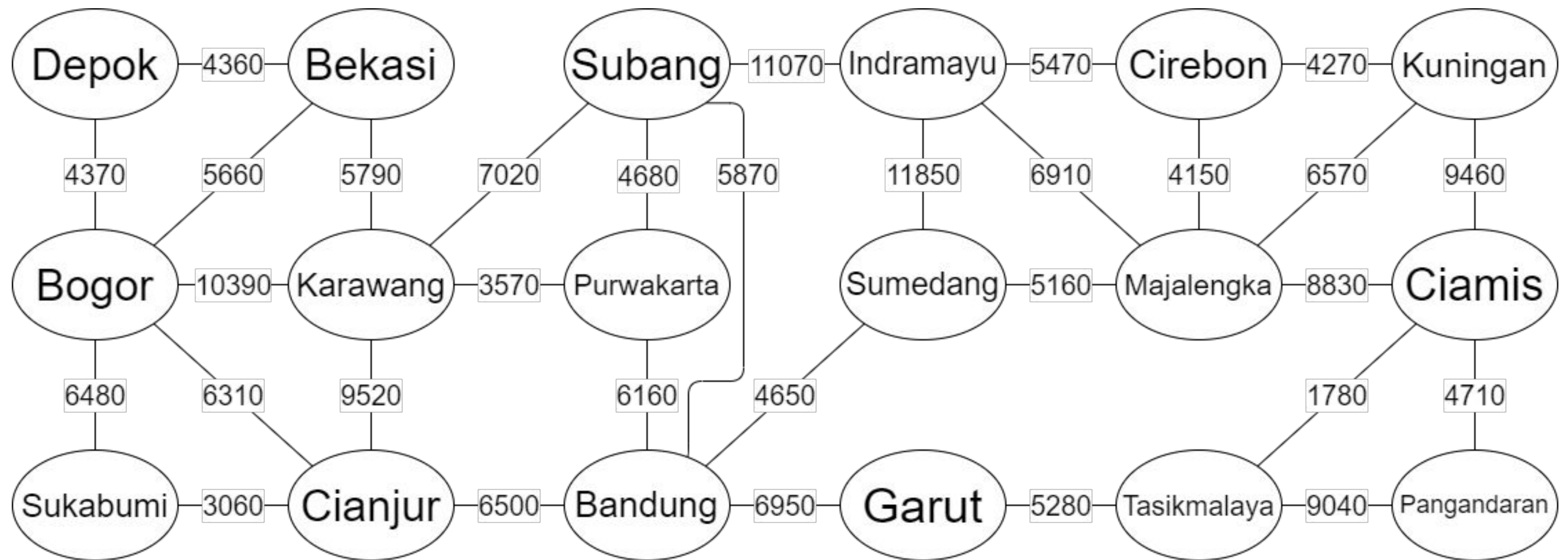


# Tugas 4

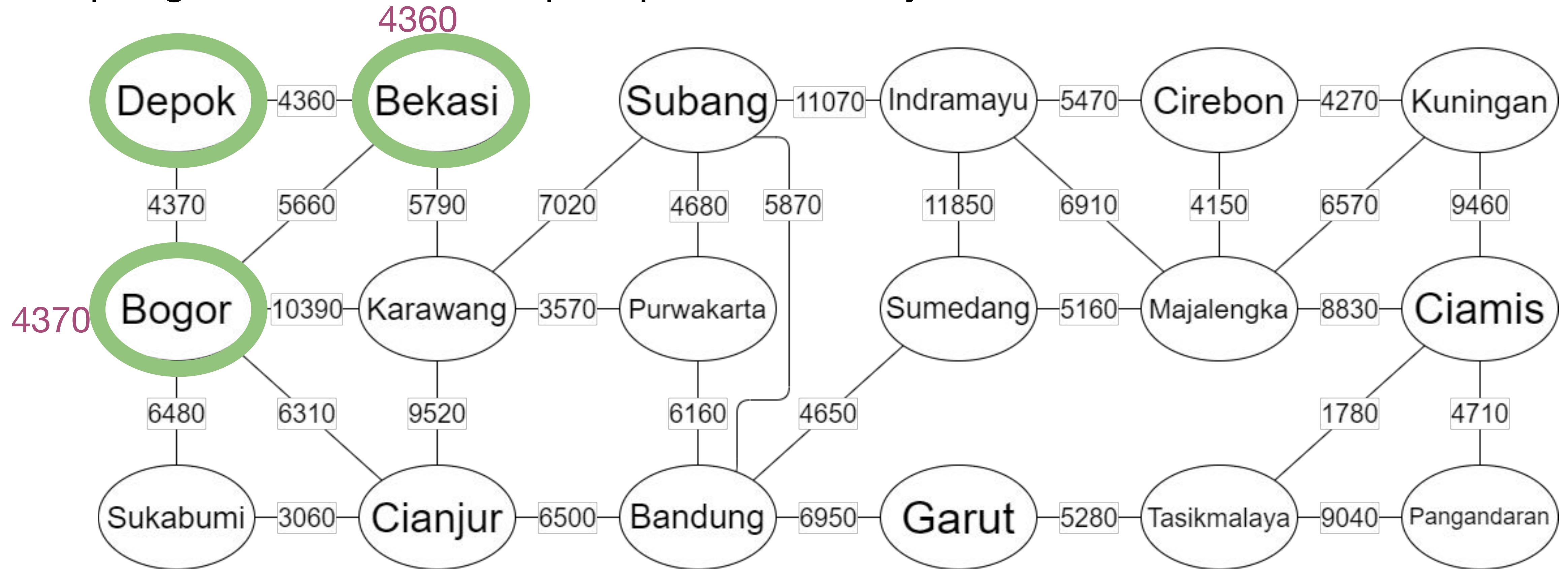
Teori Graf dan *Tree*.

1. Dengan algorithm Dijkstra, bandingkan jarak yang perlu ditempuh seseorang dari depok ke kuningan, dan dari subang ke pangandaran, sertakan pula pseudocodenya.



Data from google map

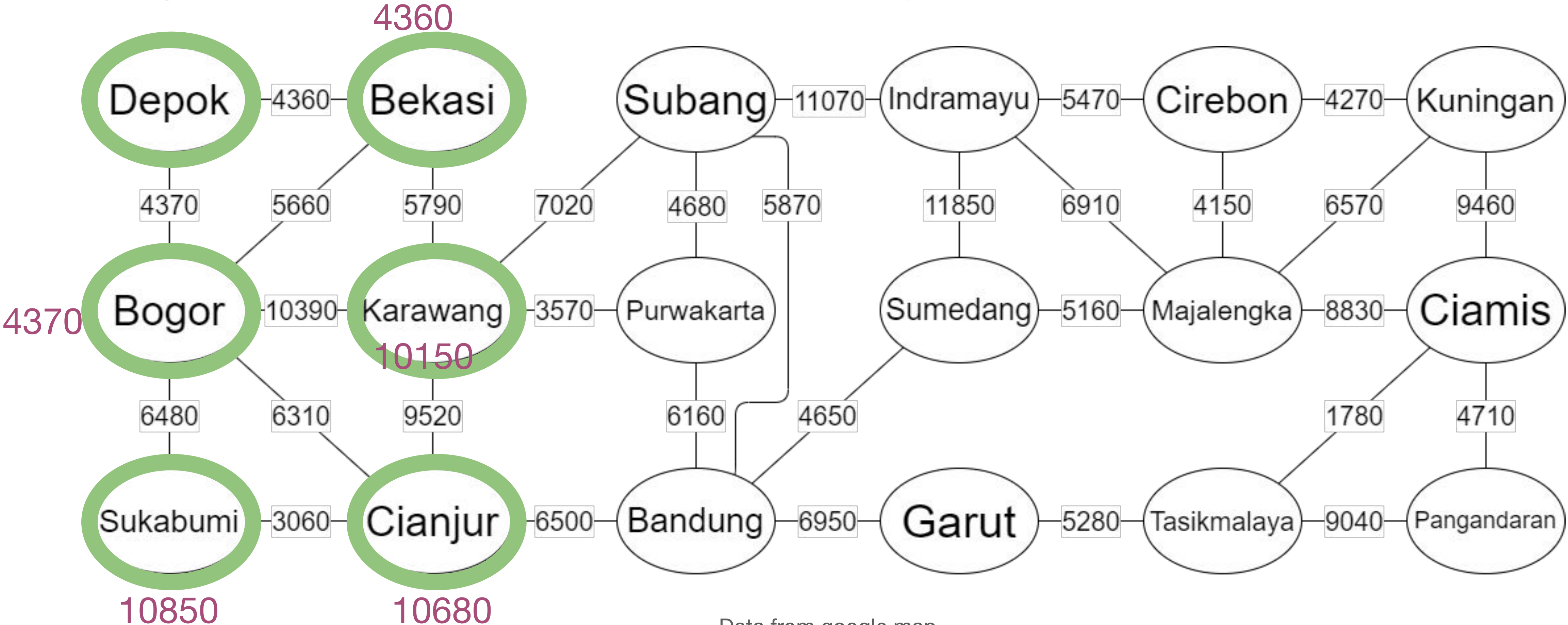
1. Dengan algorithm Dijkstra, bandingkan jarak yang perlu ditempuh seseorang dari depok ke kuningan, dan dari subang ke pangandaran, sertakan pula pseudocodenya.



Data from google map



1. Dengan algorithm Dijkstra, bandingkan jarak yang perlu ditempuh seseorang dari depok ke kuningan, dan dari subang ke pangandaran, sertakan pula pseudocodenya.

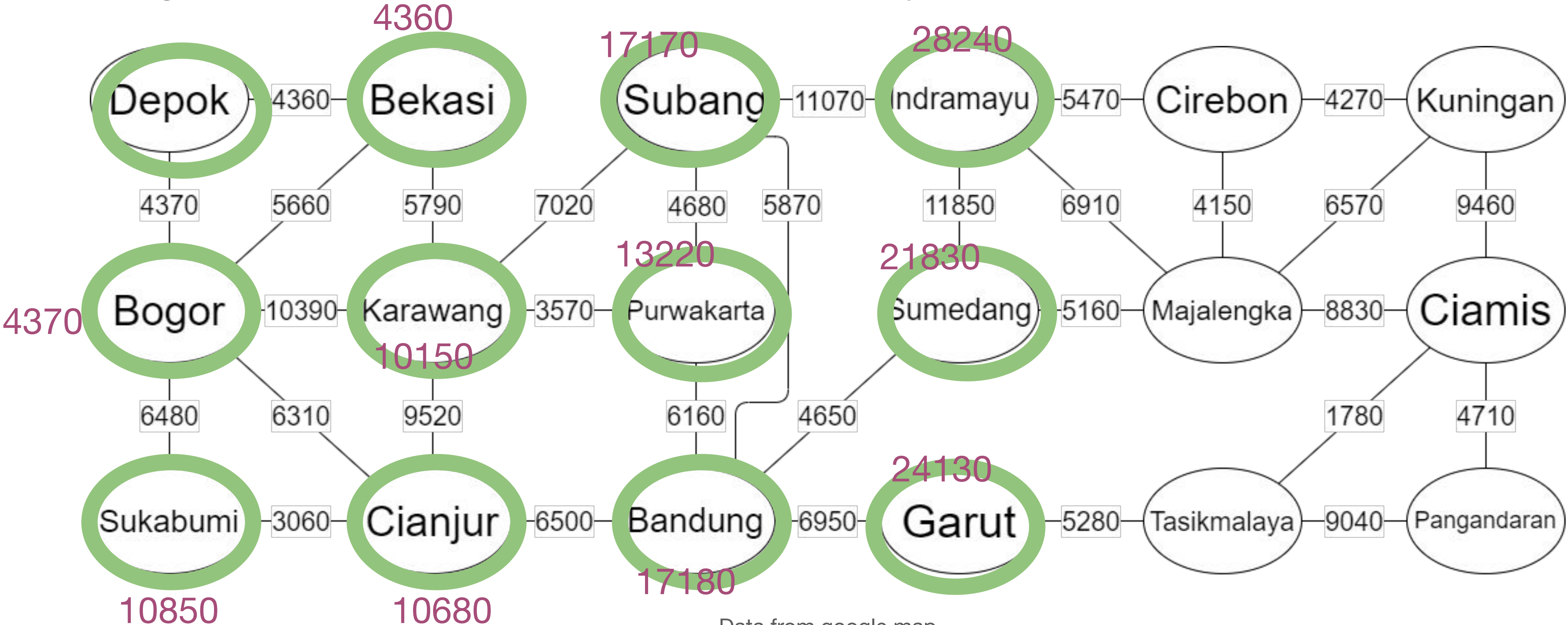


Data from google map



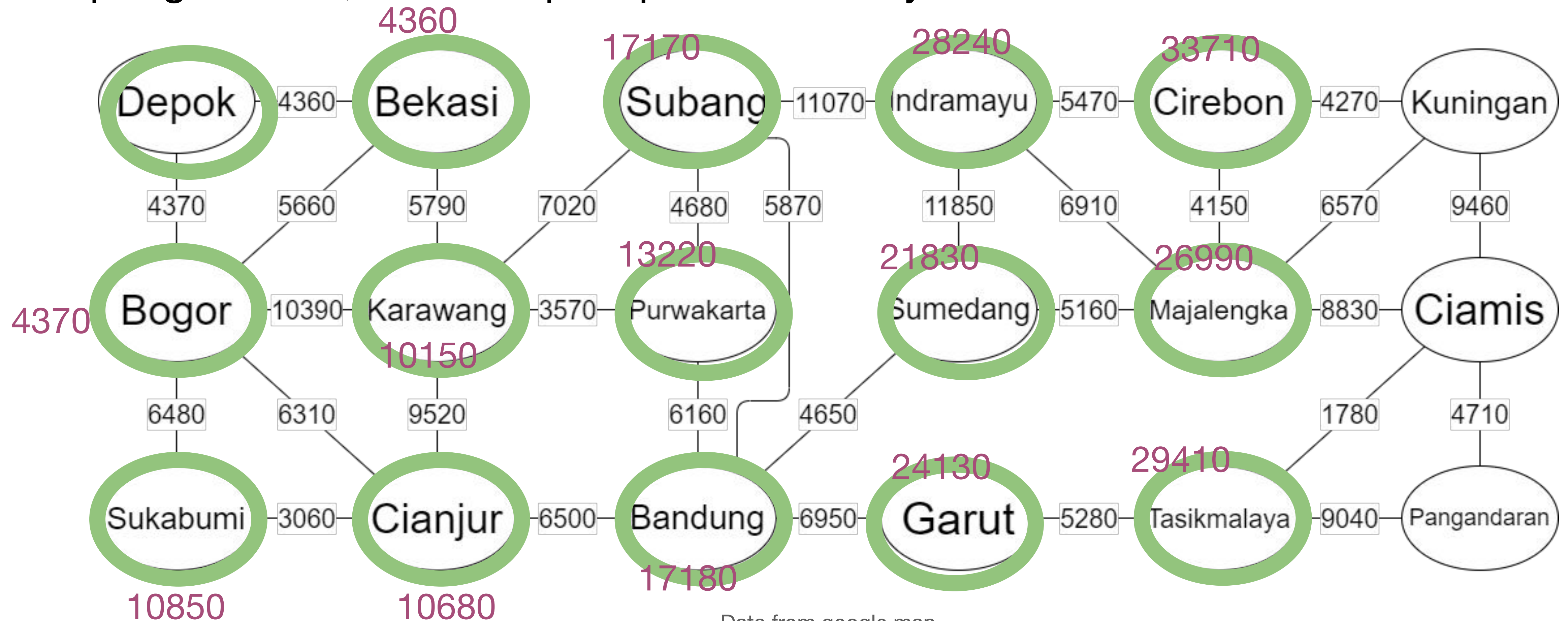


1. Dengan algorithm Dijkstra, bandingkan jarak yang perlu ditempuh seseorang dari depok ke kuningan, dan dari subang ke pangandaran, sertakan pula pseudocodenya.



Data from google map

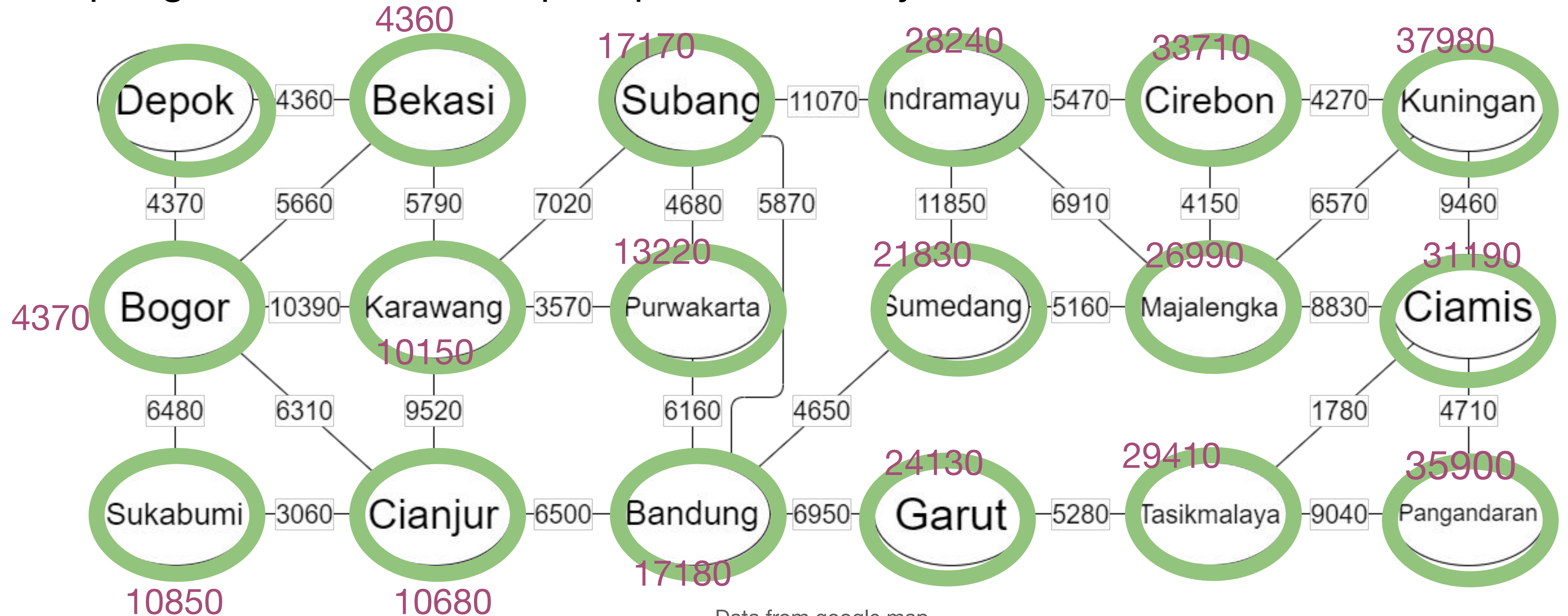
1. Dengan algorithm Dijkstra, bandingkan jarak yang perlu ditempuh seseorang dari depok ke kuningan, dan dari subang ke pangandaran, sertakan pula pseudocodenya.



## Data from google map



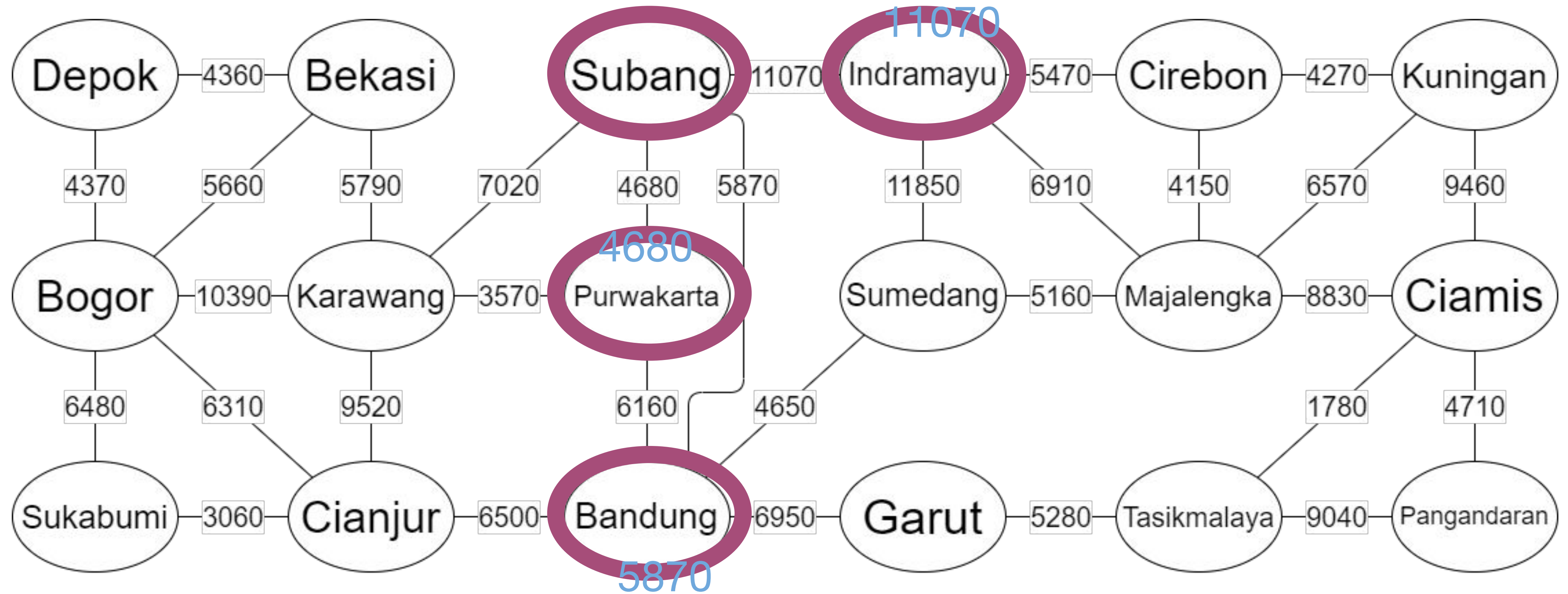
1. Dengan algorithm Dijkstra, bandingkan jarak yang perlu ditempuh seseorang dari depok ke kuningan, dan dari subang ke pangandaran, sertakan pula pseudocodenya. Jarak terdekat Depok-Kuningan: 37980



Data from google map



1. Dengan algorithm Dijkstra, bandingkan jarak yang perlu ditempuh seseorang dari depok ke kuningan, dan dari subang ke pangandaran, sertakan pula pseudocodenya.



Data from google map

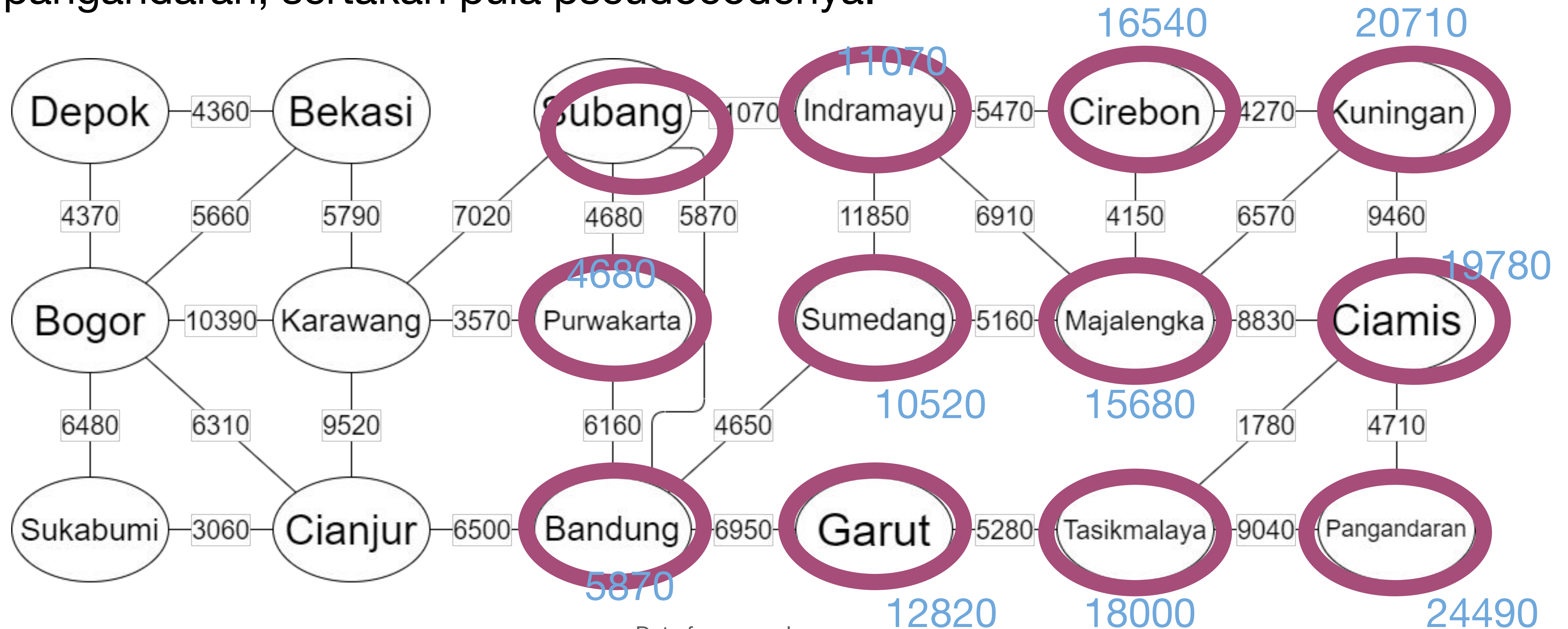






Jarak terdekat Subang-Pangandaran adalah 24490

1. Dengan algorithm Dijkstra, bandingkan jarak yang perlu ditempuh seseorang dari depok ke kuningan, dan dari subang ke pangandaran, sertakan pula pseudocodenya.



Data from google map



# Pseudocode

- 1 - Lacak semua vertices yang jarak minimumnya sudah dihitung; pada awalnya, set ini kosong (jarak minimum belum ada yang diketahui)
- 2 - Hitung jarak minimum terdekat untuk semua vertices yang bertetangga langsung dengan vertices yang sudah diketahui jarak minimumnya (pada awalnya, ini hanya vertices yang bertetangga dengan vertice sumber)
- 3 - Ulangi dan perbarui data jarak minimum hingga didapatkan vertice tujuan dan jarak minimumnya

2. Pesan yang diencode dengan bantuan tabel dibawah dirasa kurang efisien, dengan Teknik Huffman code susunlah Kembali pesan yang harus dikirim (sertakan Huffman tree nya,) tentukan pula *average bit length* nya!

10010 00100 01100 10100 00000 01111 00000  
10010 10010 10110 01110 10001 00011 01011  
10100 01010 01100 00000 01101 10011 00100  
10001 00011 01000 10001 01000 00011 00000  
10001 01000 10011 00100 01100 01111 00000  
10011 00011 00000 01101 10011 00000 01101  
00110 00110 00000 01011 01011 00000 00111  
01000 10001 01101 11000 00000

huruf	kode	huruf	kode
a	00000	n	01101
b	00001	o	01110
c	00010	p	01111
d	00011	q	10000
e	00100	r	10001
f	00101	s	10010
g	00110	t	10011
h	00111	u	10100
i	01000	v	10101
j	01001	w	10110
k	01010	x	10111
l	01011	y	11000
m	01100	z	11001



# Huffman Encoding

1 - Pesan yang harus dikirim, berdasarkan tabelnya, adalah SEMUA  
PASSWORD LUKMAN TERDIRI DARI TEMPAT DAN TANGGAL LAHIRNYA

2 - Tabel frekuensi :

A	B	C	D	E	F	G	H	I	J	K	L	M
10	-	-	4	3	-	2	1	4	-	1	3	3

N	O	P	Q	R	S	T	U	V	W	X	Y	Z
4	1	2	-	5	3	4	2	-	1	-	1	-

# Huffman Encoding

3 -