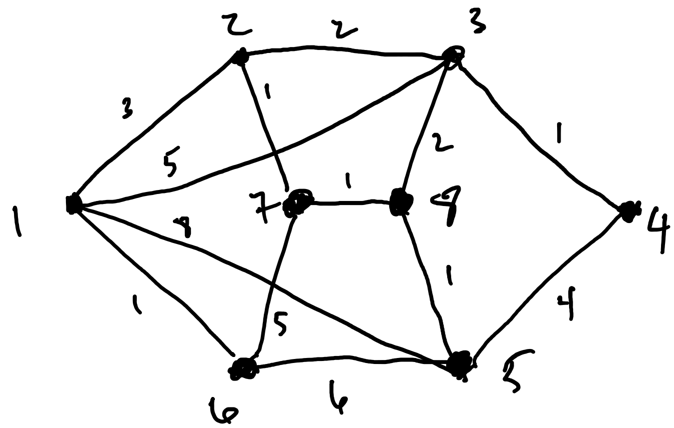
§ 7.3 # 11, 22, 28

#11



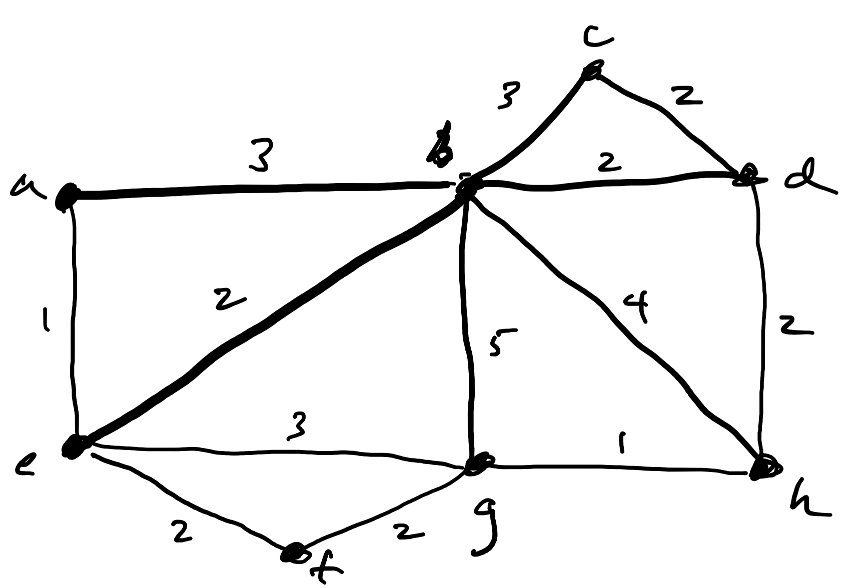
Bellman-Ford looks at paths of length 1, 2, 3, etc. until there's no change.

	1	2	3	4	5	6	7	8	
d	3	0	2	∞	∞	∞	1	∞	length 1
s	2	-	2	2	2	2	2	2	
d	3	0	2	3	11	4	1	2	length 2
s	2	-	2	3	1	1	2	7	
d	3	0	2	3	3	4	1	2	length 3
s	2	-	2	3	8	1	2	7	

$$\begin{array}{c|cccccccc} & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ \hline 1 & \infty & 3 & 5 & \infty & 8 & 1 & \infty & \infty \\ 2 & 3 & \infty & 2 & \infty & \infty & \infty & 1 & \infty \\ 3 & 5 & 2 & \infty & 1 & \infty & \infty & \infty & 2 \\ 4 & \infty & \infty & 1 & \infty & 4 & \infty & \infty & \infty \\ 5 & 8 & \infty & \infty & 4 & \infty & 6 & \infty & 1 \\ 6 & 1 & \infty & \infty & \infty & 6 & \infty & 5 & \infty \\ 7 & \infty & 1 & \infty & \infty & \infty & 5 & \infty & 1 \\ 8 & \infty & \infty & 2 & \infty & 1 & \infty & 1 & \infty \end{array}$$

No further change
(I do Dijkstra's below)

#22



1 : ae, gh

2 : bd, be, cd, dh, ef, fg

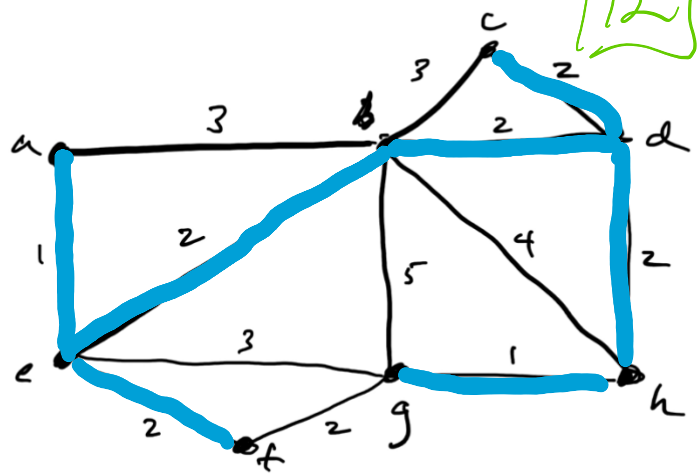
Done

3 : ab, bc, eg

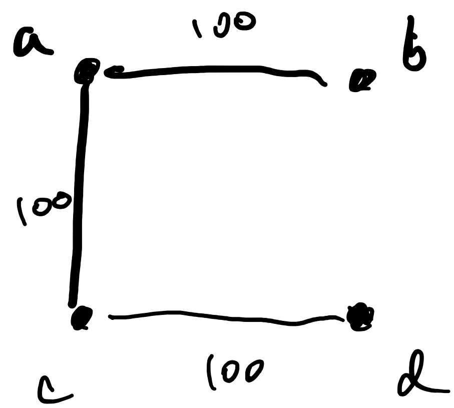
4 : bh

5 : bg

minimal weight: $\boxed{12}$

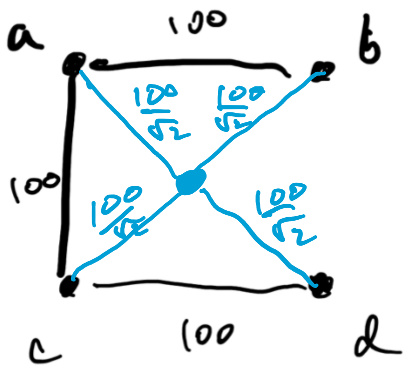


#28 a.



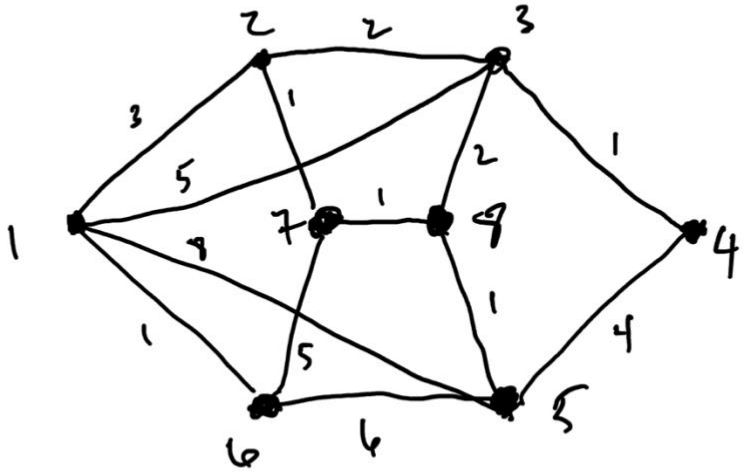
M = 300

b.



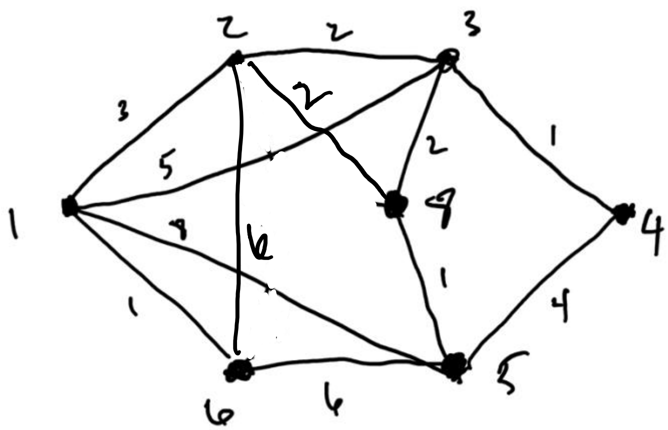
$$\frac{400}{\sqrt{2}} \approx \frac{400}{1.4} \approx 283 < 300$$

#1 2 to 5



$$
\begin{array}{c|cccccccc}
 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\
\hline
1 & \infty & 3 & 5 & \infty & 8 & 1 & \infty & \infty \\
2 & 3 & \infty & 2 & \infty & \infty & \infty & 1 & \infty \\
3 & 5 & 2 & \infty & 1 & \infty & \infty & \infty & 2 \\
4 & \infty & \infty & 1 & \infty & 4 & \infty & \infty & \infty \\
5 & 8 & \infty & \infty & 4 & \infty & 6 & \infty & 1 \\
6 & 1 & \infty & \infty & \infty & 6 & \infty & 5 & \infty \\
7 & \infty & 1 & \infty & \infty & \infty & 5 & \infty & 1 \\
8 & \infty & \infty & 2 & \infty & 1 & \infty & 1 & \infty
\end{array}
$$

I'm going to illustrate the "recursive graphs" approach: settle 7 (or need to update only nodes adjacent to 7)

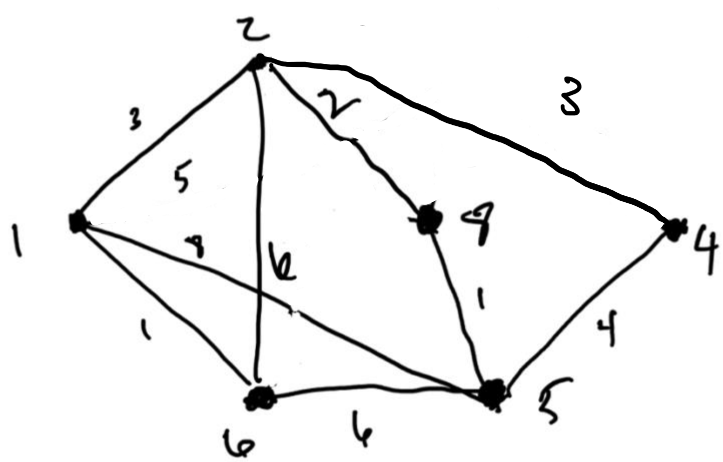


	1	2	3	4	5	6	7*	8
d	3	0	2	∞	∞	6	1	2
s	2	-	2	2	2	7	2	7

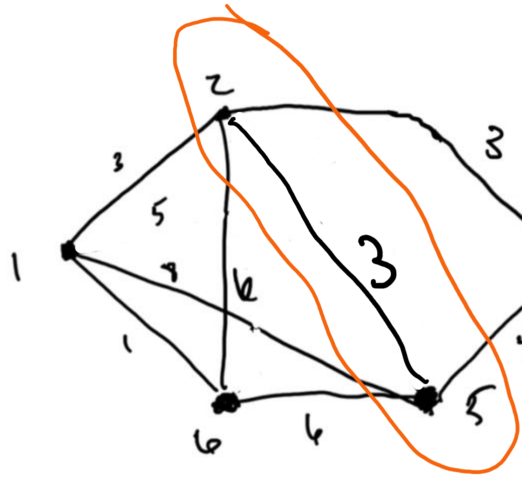
Next: settle 3 (tied w/ 8)

	1	2	3*	4	5	6	7*	8
d	3	0	2	3	∞	6	1	2
s	2	-	2	3	2	7	2	7

update only nodes adjacent to 3!



Now settle 8:



	1	2	3*	4	5	6	7*	8*
d	3	0	2	3	3	6	1	2
s	2	-	2	3	8	7	2	7

At this point a smart Dijkstra's algorithm would choose 5 to settle next, since it has minimum distance.

Path: 5 ← 8 ← 7 ← 2

Distance: $\boxed{3}$