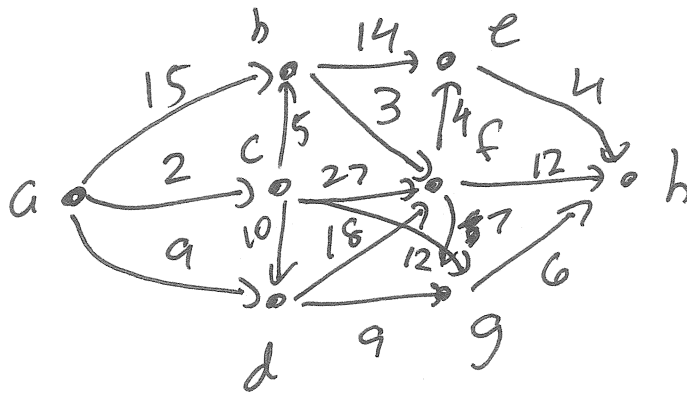


Dijkstra's Algorithm

Intro CP 6/17/26

Single Source Shortest Path Problem in a directed weighted graph.



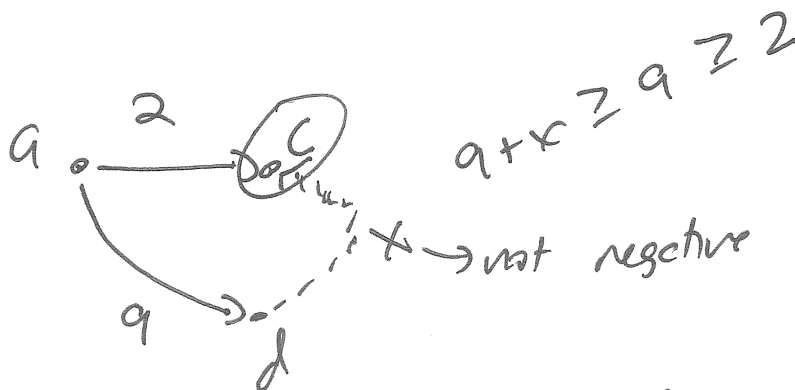
source = a

edge weights ≥ 0 .

dist

0	∞	∞	∞	∞	∞	∞	∞
a	b	c	d	e	f	g	h

priority queue of estimates (a, 0)



Intuition why
it's okay to pull
the shortest estimate
and this is optimal

Queue has to pull the shortest
current estimate!

Priority Queue

C++
pulls max element
instead of min!

	a	b	c	d	e	f	g	h
a	0	15	2	9	∞	∞	∞	∞
c	0	7/a	2	9	∞	29/a	∞	∞
b	0	7	2	9	21/c	10/c	∞	∞

edge (f, 10)

a → b 15 $\text{dist}[b] = \text{dist}[a] + 15 = 15$

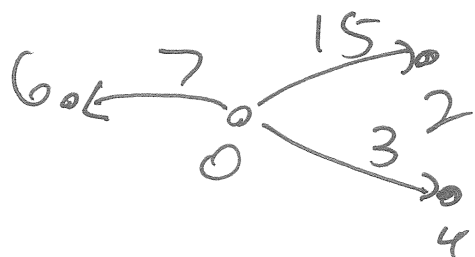
a → c 2 $\text{dist}[c] = \text{dist}[a] + 2 = 2$

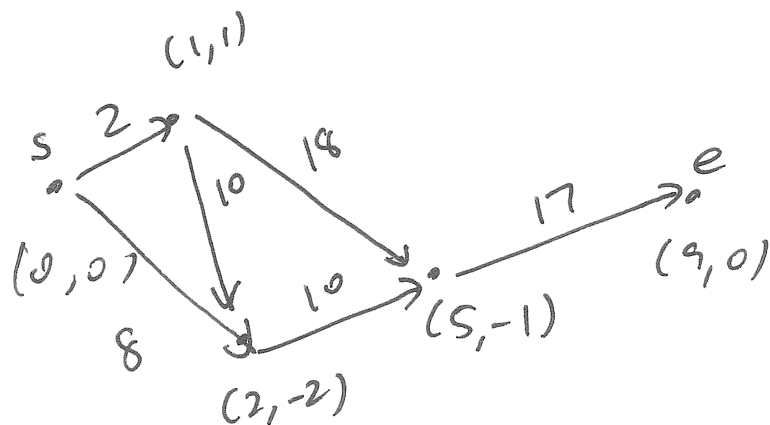
a → d 9 $\text{dist}[d] = \text{dist}[a] + 9 = 9$

d	0	7	2	9	21	10	18	∞
f	0	7	2	9	14	10	17	22
e	0	7	2	9	14	10	17	20 18
g	0	7	2	9	14	10	17	<u>18</u>
g	0	7	2	9	14	10	17	18
h	0	7	2	9	14	10	17	18

Rm time $E \lg E$, $E \leq V^2$
 $O(E \lg V)$ $\lg V^2 = \underline{2 \lg V}$

graph[0] → $\left[\overset{to}{(2, 15)}, \overset{w}{(4, 3)}, (6, 7) \right]$





Complete Graph (Build Before)

OR compute edges as you go + never fully build it

if $(\text{dist}[\text{old } V] + w < \text{dist}[\text{new } V]) \{$
 $\text{dist}[\text{new } V] = \text{dist}[\text{old } V] + w;$
 $\text{prev}[\text{new } V] = \text{old } V;$
 $\}$

	a	(b)	(c)	d	(e)	(f)	g	(h)
prev	a	c	a	a	(f)	(b)	f	(e)

Path a → h a → c → b → f → e → h

ans = ~~path~~ = [end]

while (end != start) {

end = path[end]

ans = end + ans

}

start ans whole path