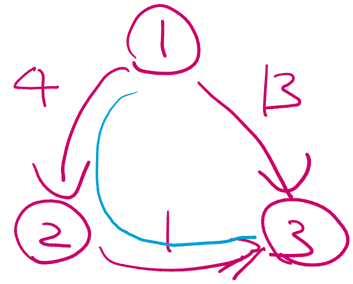
07-27

Monday, July 27, 2020 6:31 PM

SSSP [Single Source Shortest Path]

- $G = (V, E)$
- $e \in E$, weight(e) / cost(e)
- Source vertex s



want to compute shortest distance/path

$s \rightsquigarrow u \quad \forall u \in V$

---

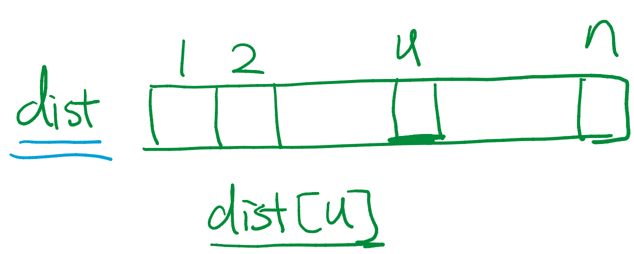
① Is G a DAG?

② $cost(e) \geq 0 \; \forall e \in E$

	Yes	No
Yes	SP on DAG	Dijkstra's algo $O(n^2)$ $O(m \lg n)$
No	$O(n+m)$	Bellman Ford $O(mn)$

SP on DAG

$s \rightsquigarrow u \xrightarrow{cost[u,v]} v$



relax(u, v) {
 if (dist[u] + cost[u,v] < dist[v]) {
 dist[v] = dist[u] + cost[u,v];

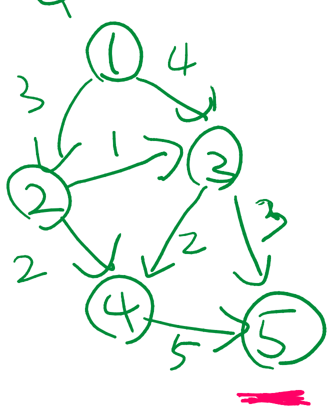
```
        dist[v] = dist[u] + cost[u, v];
        π[v] = u;
    }
}
```

SP on DAG

```
for (u in topological_sort(G)) {
    for (v in G.neighbor(u)) {
        relax(u, v);
    }
}
```

$O(n+m)$

G

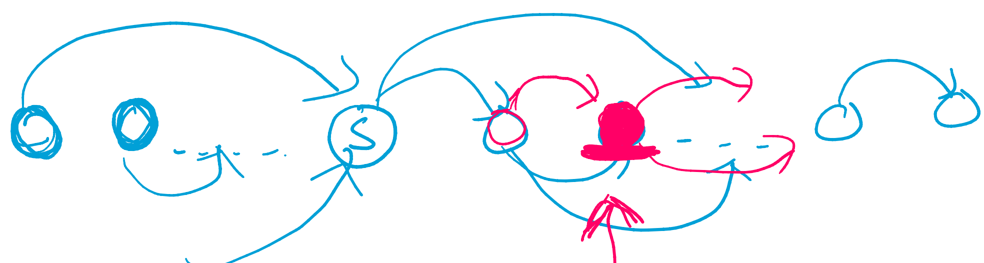


$s = 2$

1, 2, 3, 4, 5

dist

1	2	3	4	5
∞	0	1	2	4



dist[·]

(A) make sense

(B) make no sense

Dijkstra's algo

$N = \{ \}$

$V = \{1 \ldots n\}$

$N = \{ \}$ $V = \{1 \dots n\}$

while ($\boxed{N} \neq V$) {   min heap [only keeps unpicked vertices]

① { pick vertex $u \in V \setminus N$ with the smallest dist[·] value

$O(1)$ $O(n)$

$O(\lg n)$ $O(1)$ ② $N = N \cup \{u\}$

$O(\text{degree}(u) \cdot \lg n)$ $O(\text{degree}(u))$ ③ for (v in G.neighbor(u)) {

relax (u, $\boxed{v}$);

}

bubble up

$O(m \lg n)$

G

$\boxed{s=2}$

	1	2	3	4	5
	×	×	×	×	×
dist	3	0	1	5	4

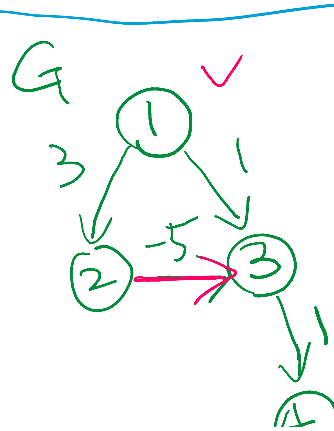
$N = \{2, 3, 1, 5, 4\}$

$u =$

	1	2	3	4	5
picked	F	F	F	F	F

$O(n^2)$

G



$s = 1$

	1	2	3	4
	×	×	×	×
dist	0	3	1	2
			-2	

y'
(4)

→ ~~~

$O((n+m)\lg n)$ $O(m \lg n)$

Bellman-Ford

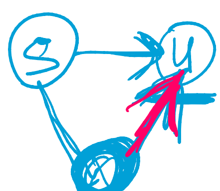
```
B.F.(G, s) {        dist[s] = 0;
    for( i = 1, 2, ... , n-1) {
        for( u = 1, 2, ... , n) {
            for (v in G.neighbors(u)) {
                relax(u, v);
            }
        }
    }
    return dist[];
}
```

$O(nm)$

$i = k$

$s \rightsquigarrow u$ # edge $= k$

dist[u]



(A) make sense

(B) make no sense

$s \rightsquigarrow u$

cost (path $s \rightsquigarrow u$)

num_edges(path $s \rightsquigarrow u$)

$s \xrightarrow{3} \circ \xrightarrow{5} \circ \xrightarrow{4} u$

cost is 12

edge is 3

$s \rightsquigarrow u$

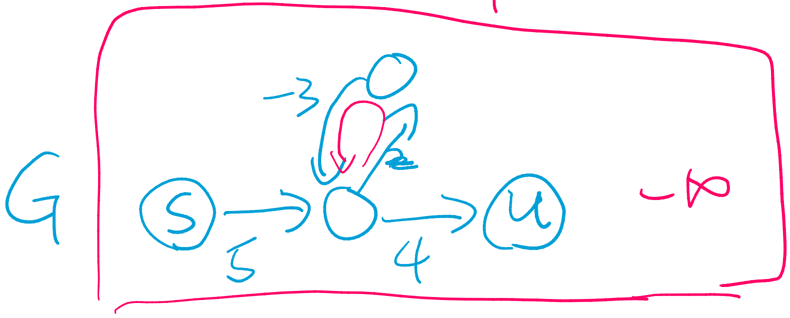
most # edges

$s \rightsquigarrow u$

shortest path

most # edges

$|V|-1$ $\quad$ $n-1$



negatively weighted cycles ✗

shortest distance is not well define

$\boxed{\text{simple path}}$ ✓ $S \to 1 \to 2 \to u$

✗ $S \to 1 \to 2 \to 1 \to 2 \to 1 \to u$

```
B.F.(G, S) {
    for (i=1... n-1) {
        for ( (u,v) ∈ E) {
            relax (u,v)
        }
    }
```

```
for ( (u,v) ∈ E) {
    relax(u,v);
}
```

dist[ ]

}

}

shortest distance

dist[ ]

shortest path ??

$s \to \cdots \to u \to v$

$\pi[v] = u$

APSP [all pair shortest path]

- $G = (V, E)$
- $e \in E$, $cost(e)$

$$\begin{array}{c|ccc} & 1 & v & n \\ \hline 1 & & & \\ u & & & \\ n & & & \end{array}$$

want short distance $dist[u,v]$ $\forall u, v \in V$

run SSSP n times

① Is G a DAG

② all edge costs are positive

	Y	N
Y	SP on DAG	Dijkstra

Dijkstra: $n^2 \lg n$ $O(n^3)$ or $O(nm \lg n)$ $\sim n^3$

$m = O(n^2)$

$m = \Omega(n)$

	SP on DAG	$n^2 \lg n \sim n^3$ or $O(nm\lg n)$
N	$O(nm)$	Bellman Ford $O(n^2 m)$
	$n^2 \sim n^3$	$n^3 \sim n^4$ ↳ $O(n^3)$

$\Longrightarrow$ **Floyd-Warshall**

$V = \{1, 2, 3, \dots, n\}$

$\boxed{f^k(u,v)}$ = shortest distance from u to v w/ vertice on shortest path have to be in $\{1, 2, \dots, k\}$

$u \to 1 \to v$

$f^1(u,v)$ $\{1\}$

$f^2(u,v)$ $\{1,2\}$

$f^n(u,v)$

$u \to 3 \to v$

X $f^1(u,v)$ $\{1\}$

x $f^2(u,v)$ $\{1,2\}$

✓ $f^3(u,v)$ $\{1,2,3\}$

$f^n(u,v)$

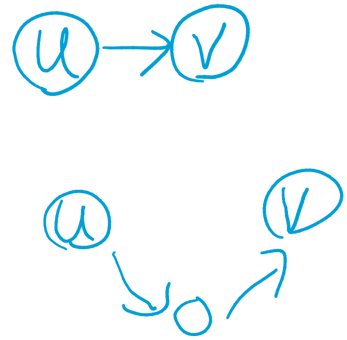
$u \rightsquigarrow v$

$\{1, 2, \dots, n\}$

$$f^0(u,v) = \begin{cases} \text{cost}[u,v] \\ \infty \end{cases}$$

$\Downarrow$

adj. matrix



$$f^0(u,v) \Rightarrow f^1(u,v) \Rightarrow f^2(u,v) \Rightarrow \cdots \Rightarrow f^n(u,v)$$

base case

goal

$$\boxed{f^k(u,v) = \min\{f^{k-1}(u,v),\ f^{k-1}(u,k) + f^{k-1}(k,v)\}}$$

$\{1, 2, \ldots, k-1, k\}$

$f^{k-1}(u,v)$

$f^{k-2}(u,v)$

$\vdots$

$f^0(u,v)$

$\boxed{f^{k-1}(u,v)}$

u v

don't use k

$f^{k-1}(u,k)$

k

$f^{k-1}(k,v)$

use k

f^0 f^1 f^2

$$\boxed{f^{k-1}(u,k) + f^{k-1}(k,v)}$$

F.W(G) {

$f^0 = \text{cost}[u,v]$

$f^1\ f^2\ f^3 \ldots f^n$

for (k = 1, 2, ..., n) {

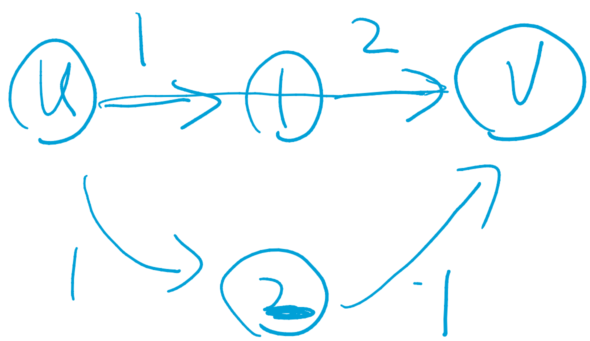
for (u = 1.. n) {

$O(n^3)$

```
for (u = 1 .... n) {
    for (v = 1 .... n) {
        A[u,v] = min{A[u,v], A[u,k] + A[k,v]}
    }
}
return A;
}
```

$f^1(u,v) = 3$

$\{1\}$



$f^2(u,v)$