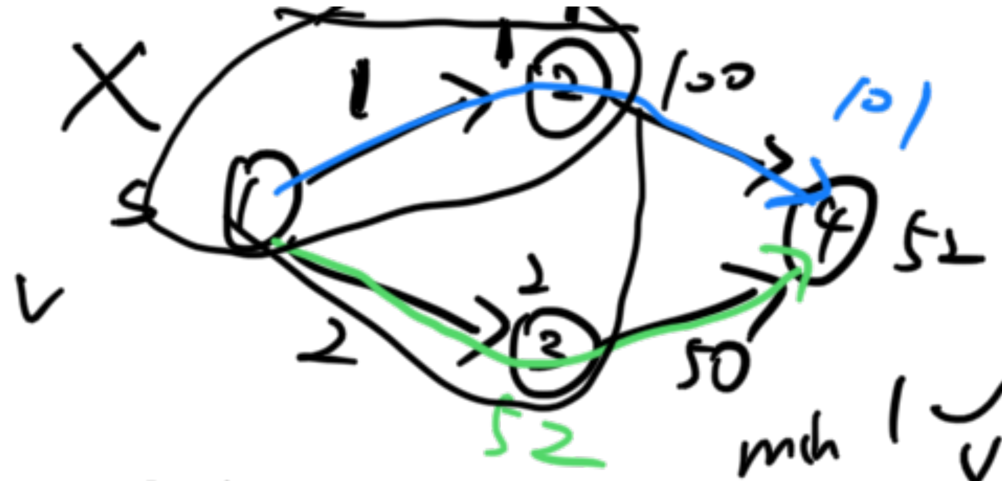
4.4. Shortest Path

Single Source shortest path



$G = (V, E)$

$\ell(e)$: cost of each edge. $\ell((1,2)) = 1$
$\ell((1,3)) = 2$

min $1 \underset{X}{\rightsquigarrow} 4 = 101$

· Goal: find shortest path from s to all other nodes.

Dijkstra's alg:

X: explored nodes.
(shortest paths are known)

$d[u]$ cost of min path from s to u via X

$s \underset{X}{\rightsquigarrow} u$

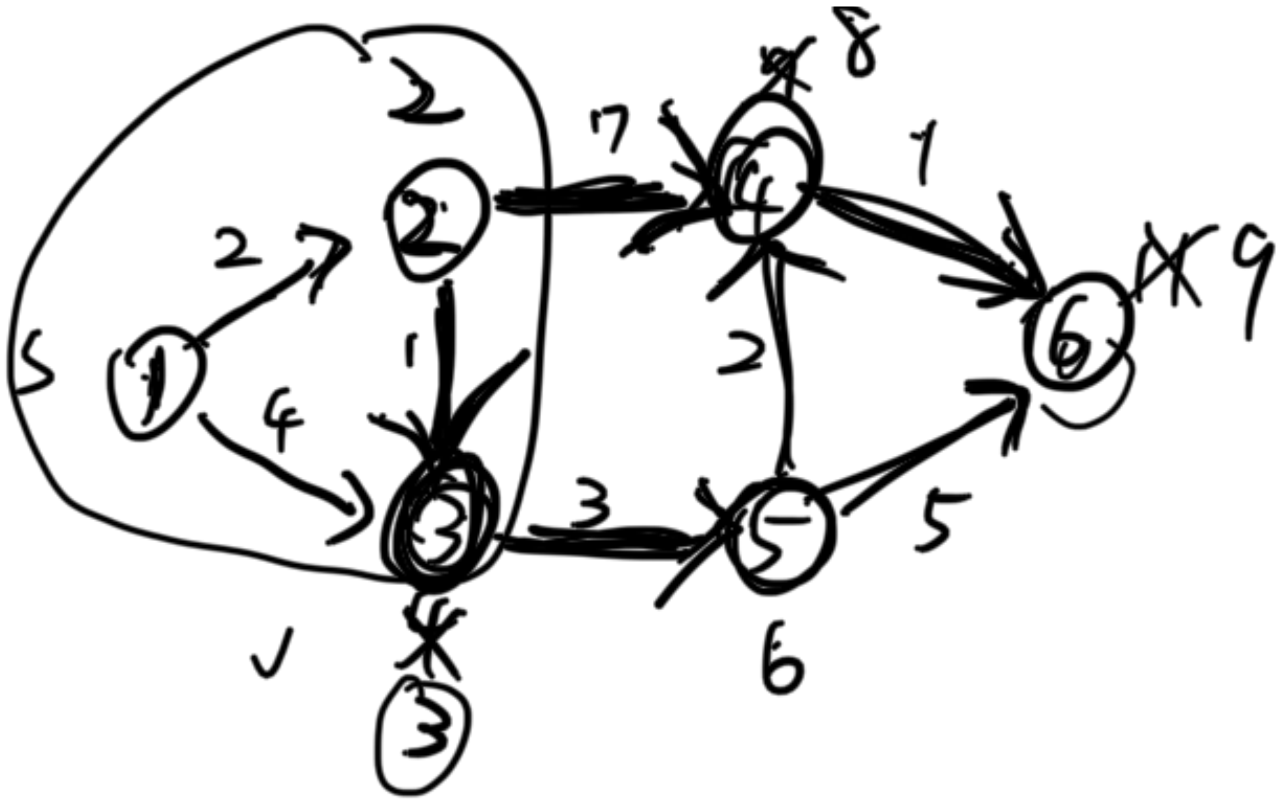
To increase X

① Find u with min $d[u]$

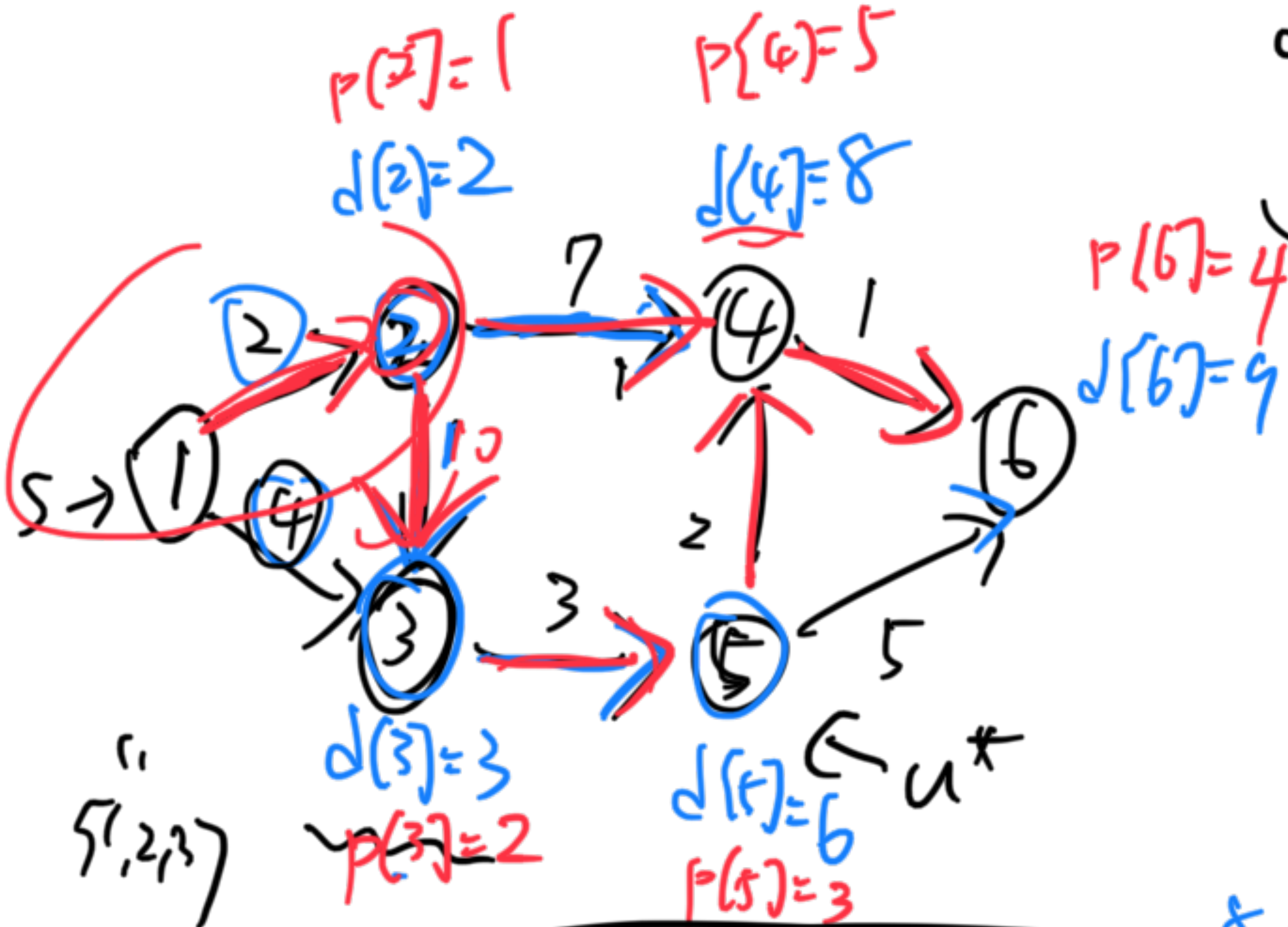
② Add u to X

$\forall v$ in neighbor(u)

$d[v] = \min\{d[v], d[u] + \ell(u,v)\}$



	$X=\{1\}$	$X=\{1,2\}$	$X=\{1,2,3\}$	$X=\{1,2,3,5\}$
d[1]	0			
d[2]	2			
d[3]	4 →	3		
d[4]	∞ →	9	9 →	8
d[5]	∞	∞ →	6	
d[6]	∞	∞	∞ → 11 →	9
	pick 2	pick 3	pick 6	pick 4



{1,2,3}

$V-X=\{4,5,6\}$

$\underline{X}$: set of nodes with known shortest path.

$\boxed{\begin{array}{l} u \text{ is with smallest } d(u) \text{ value} \\ \Rightarrow d(u) \text{ is the shortest } s \rightsquigarrow u \text{ path.} \end{array}}$

Dijksta's alg:

Initial $X=\{s\}$, $d[u] = \begin{cases} e(s,u) & \text{if } (s,u)\in E \\ \infty & \text{otherwise} \end{cases}$

p[u]=s

Heap $Q = \{d[u]\ \forall u \in V\}$

for $i = 1 \sim n-1$

$u^* \leftarrow \boxed{\arg\min_{u \in V-X} d[u]}$ $\leftarrow$ $Q.pop()$ ... $O(\log n)$

infinites.

$Q.updatevalue$ ... $O(\log n)$

$O(\text{outdegree}(u^*))$ $\Big\{$ for each v with $(u^*, v) \in E$

if $d[u^*] + \ell(u^*, v) < d[v]$: $d[v] = d[u^*] + \ell(u^*, v)$.

$\boxed{p[v] = u^*}$

add u^* to X.

$\in V - X$

Thm: u is the unexplored node with min d

$\Rightarrow d[u]$ is the shortest path $s \rightsquigarrow u$.

pf: $d[u]$: shortest $s \rightsquigarrow u$ path through X.

for any other path P $s \rightsquigarrow u$ through another node $w \in V-X$

$P: s \rightsquigarrow y \to w \rightsquigarrow u$

$s \rightsquigarrow y \in X$, $w \in V-X$

$\ell(P) = \ell(s \rightsquigarrow y) + \ell(y \to w) + \ell(w \rightsquigarrow u)$

$\geq \ell(s \rightsquigarrow y) + \ell(y \to w) \geq d[w]$

min $s \rightsquigarrow w$ path via X.

since $d(u)$ is min $\Rightarrow d[u] \leq d(w)$

$\leq \ell(P)$

Implementation:

- Heap to store $d(u)\ \forall u \in V-X$.

  n pops. m update value. n push.

  totally $O(n\log n + m\log n + n\log n)$

  $= O(m\log n)$

- Use Fibonacci heap $\Rightarrow O(n\log n + m + n)$

  $= O(n\log n + m)$.

  $O(1)$ push

  $O(\log n)$ pop

  $O(1)$ update value