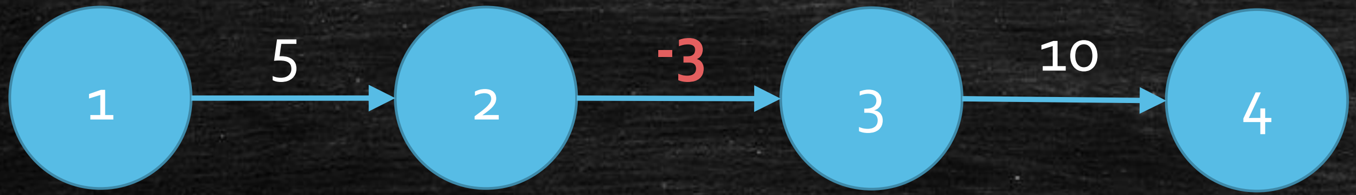
Shortest Path (Negative)

Bellman-Ford

Is Dijkstra Algorithm always correct?

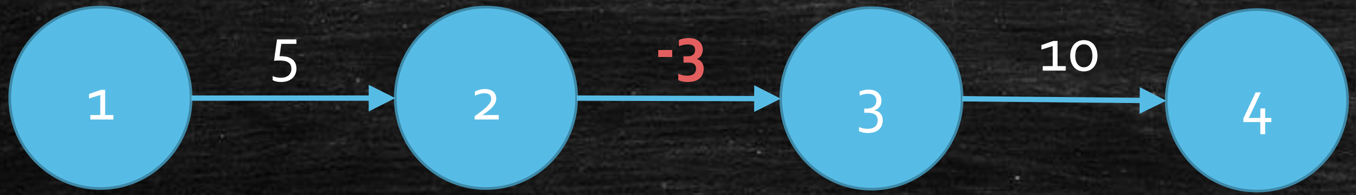
Shortest Path with Negative Length

- What if edges may have **negative** weight?
- Distance: $5 - 3 + 10 = 12$



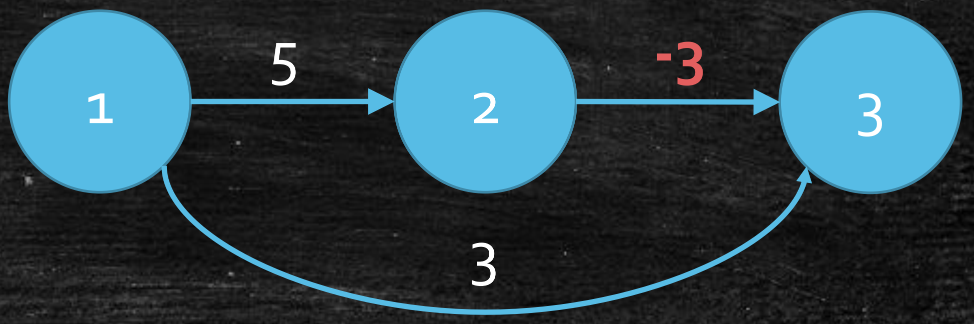
Shortest Path with Negative Length

- What if edges may have **negative** weight?
- Distance: $5 - 3 + 10 = 12$



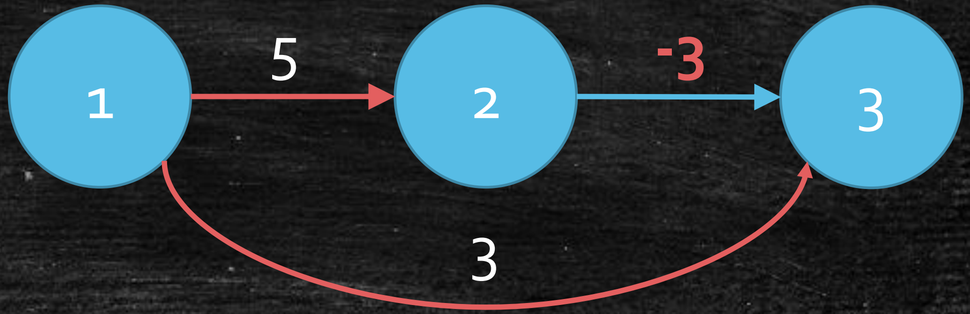
Can we still use Dijkstra?

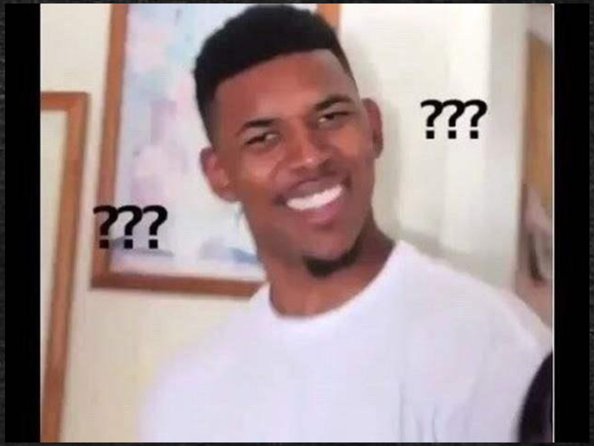
- Try Dijkstra on this small graph?



Can we still use Dijkstra?

- Try Dijkstra on this small graph?
- The Fake **SPT** we get
- It is not True **SPT** because
  - $dist_T(3) = 3 > dist(3) = 5 - 3 = 2$

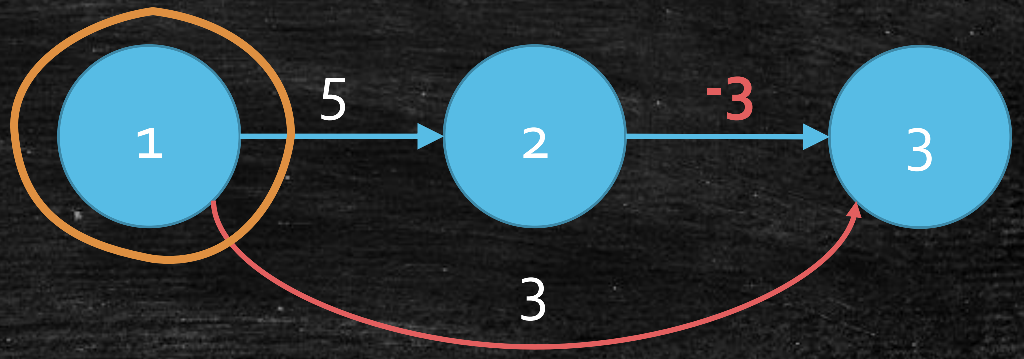




But we have proved it???

What we have proved (last lecture)

- We can explore an **SPT**.
- Choose the closest vertex.
- {1} is an **SPT.**
- 3 is the closest vertex.

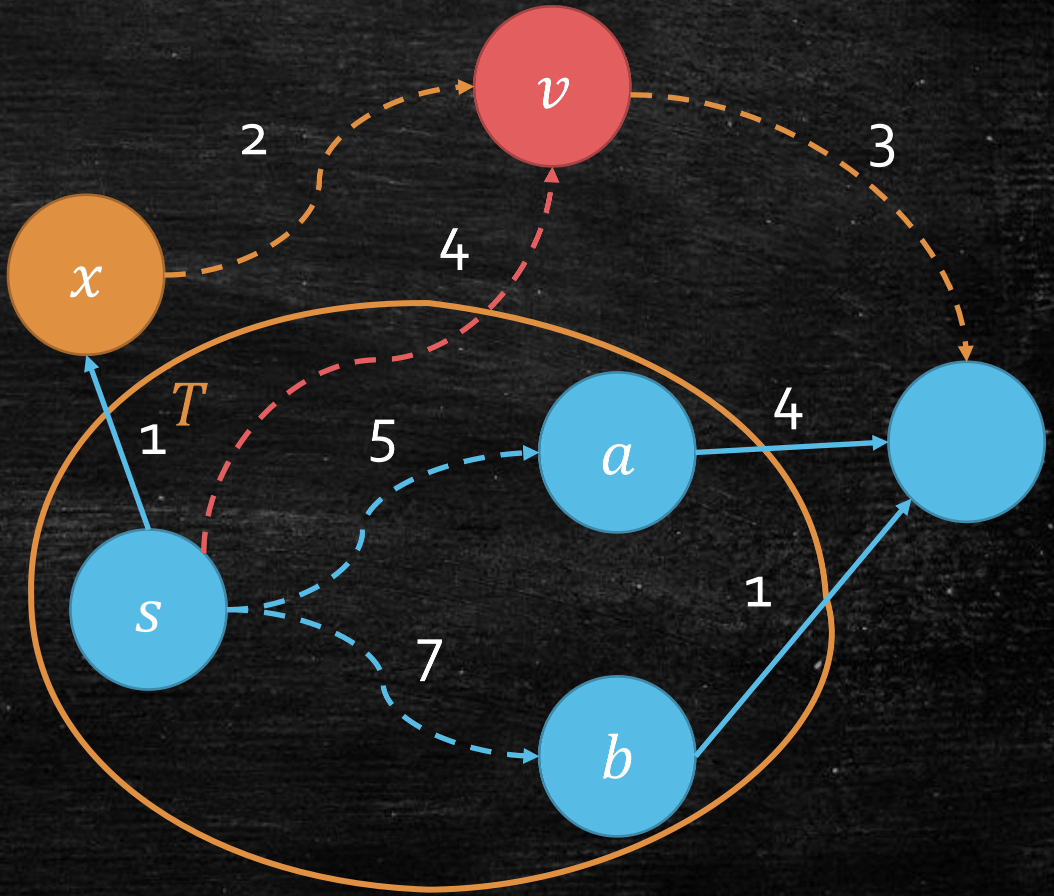


- We should have something wrong in the proof!

Go back to the proof!

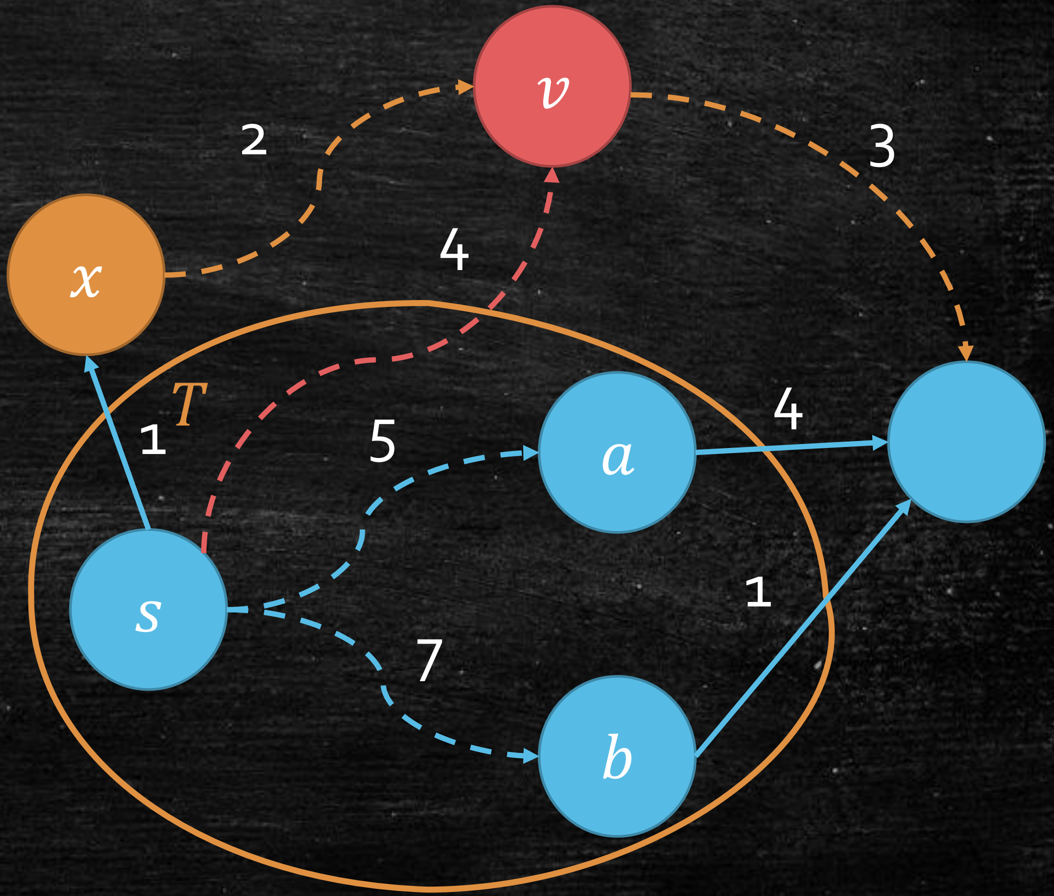
Prove $dist_T(v) \le dist(v)$ **AGAIN**!

- Try to explore v into T
- Naturally, we should connect it to $\underset{u \in T}{\operatorname{argmin}}\, dist_T(u)$
- Assume $dist_T(v) > dist(v)$
- $x \notin T,\ s \to x \to v < dist_T(v)$
- $dist_T(x)$ is a part of $s \to x \to v$
- $dist_T(x) < dist_T(v)$
- **Contradiction!**



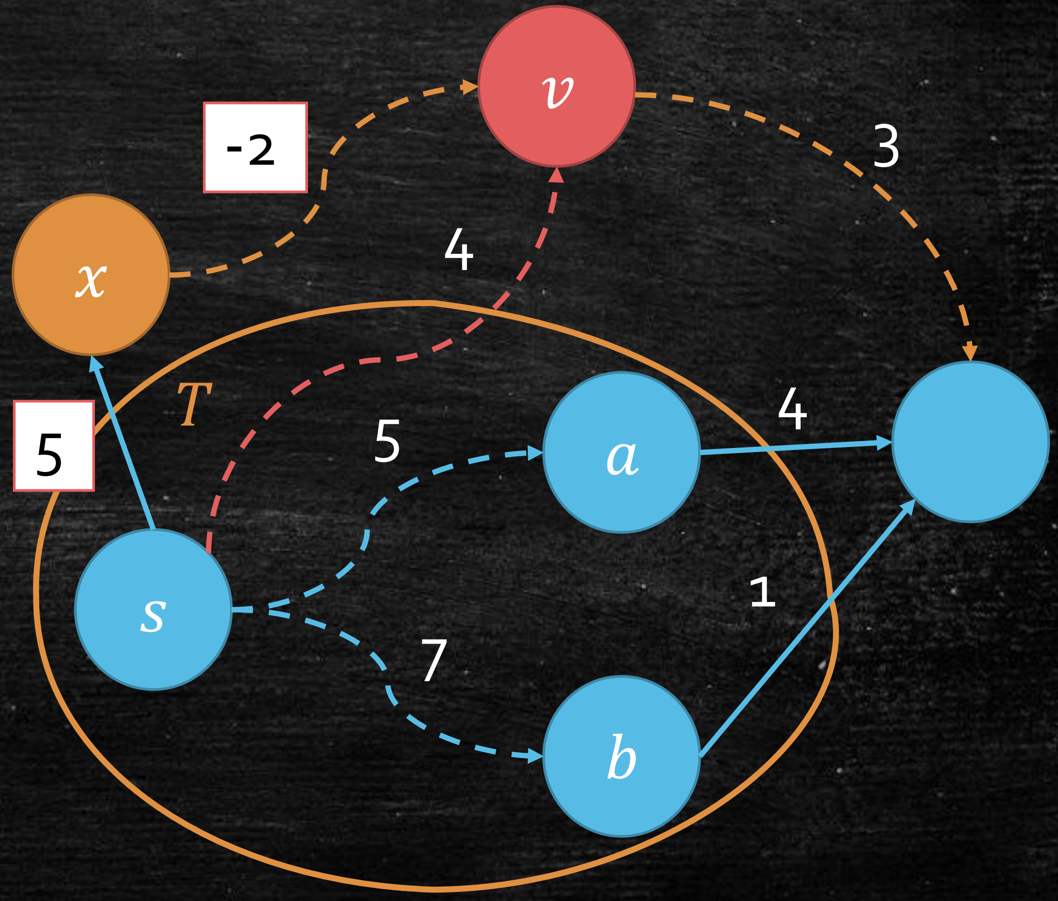
Prove $dist_T(v) \leq dist(v)$ **AGAIN**!

- Try to explore v into T
- Naturally, we should connect it to $\underset{u \in T}{\operatorname{argmin}} \, dist_T(u)$
- Assume $dist_T(v) > dist(v)$
- $x \notin T,\ s \to x \to v < dist_T(v)$
- $dist_T(x)$ is a part of $s \to x \to v$
- $dist_T(x) < dist_T(v)$
- **Contradiction!**



Prove $dist_T(v) \le dist(v)$ **AGAIN**!

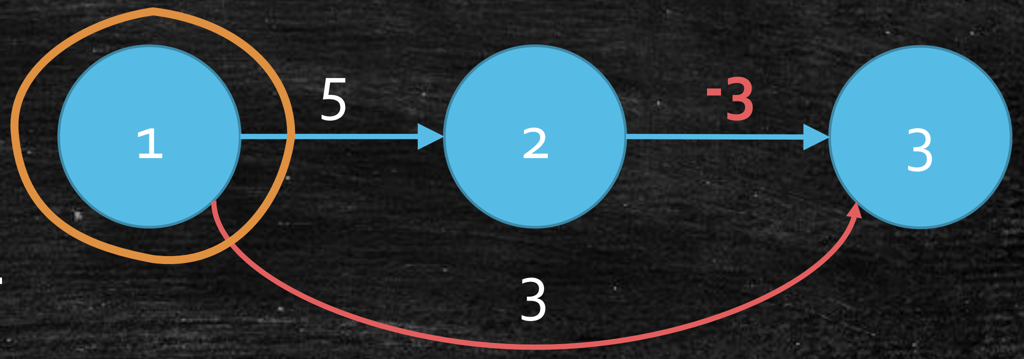
- Try to explore v into T
- Naturally, we should connect it to $\underset{u \in T}{\operatorname{argmin}}\, dist_T(u)$
- Assume $dist_T(v) > dist(v)$
- $x \notin T,\ s \to x \to v < dist_T(v)$
- $dist_T(x)$ is a part of $s \to x \to v$
- $dist_T(x) < dist_T(v)$
- **Contradiction!**



New solution Bellman-Ford!

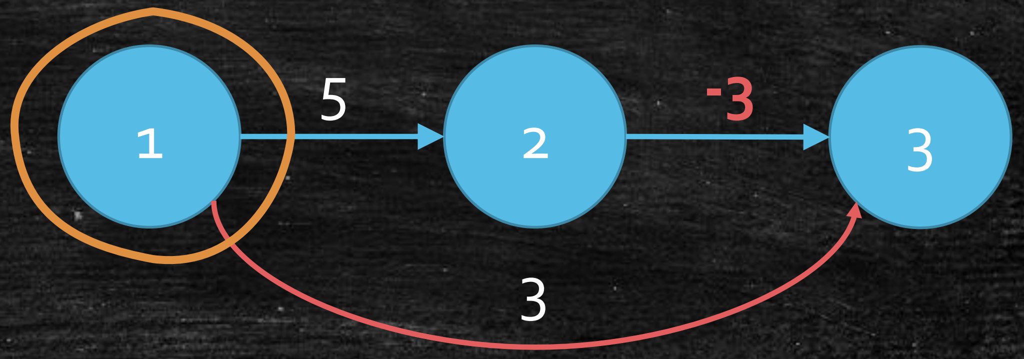
Another view of the problem

- Dijkstra
  - If we update 3 into **SPT**,
  - $dist(3)$ needn't be updated any more!
  - We only need to update others!
- Now
  - It is not correct.
  - It can be updated by $dist(2) - 3$.
- Simply solution
  - Don't chose vertex 3.
  - Keep updating everyone!



Another view of the problem

- Conclusion
- Dijkstra is very clever
  - It follows a clever **order**
  - Each edge can be only used **once** in updating.
- Now the order is **not true**
- We can only be **stupid**.



Bellman-Ford

Bellman-Ford

Function bellman_ford(G, s)

$dist[s] = 0, dist[x] = \infty$ for other $x \in V$

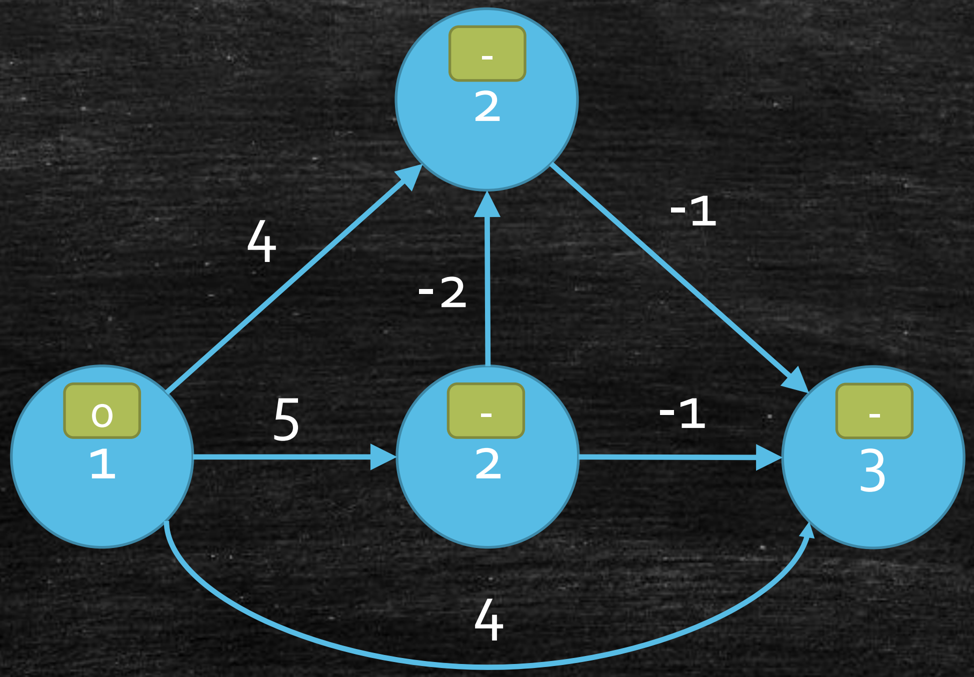
while $\exists dist[x]$ is updated

for each $(u, v) \in E$

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

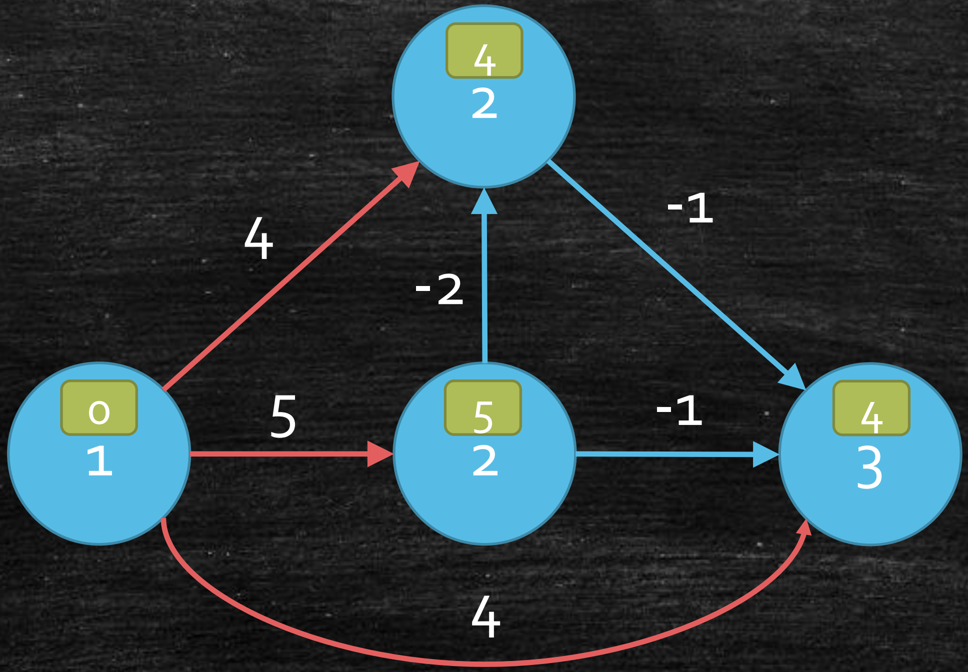
Sample run

Round 1



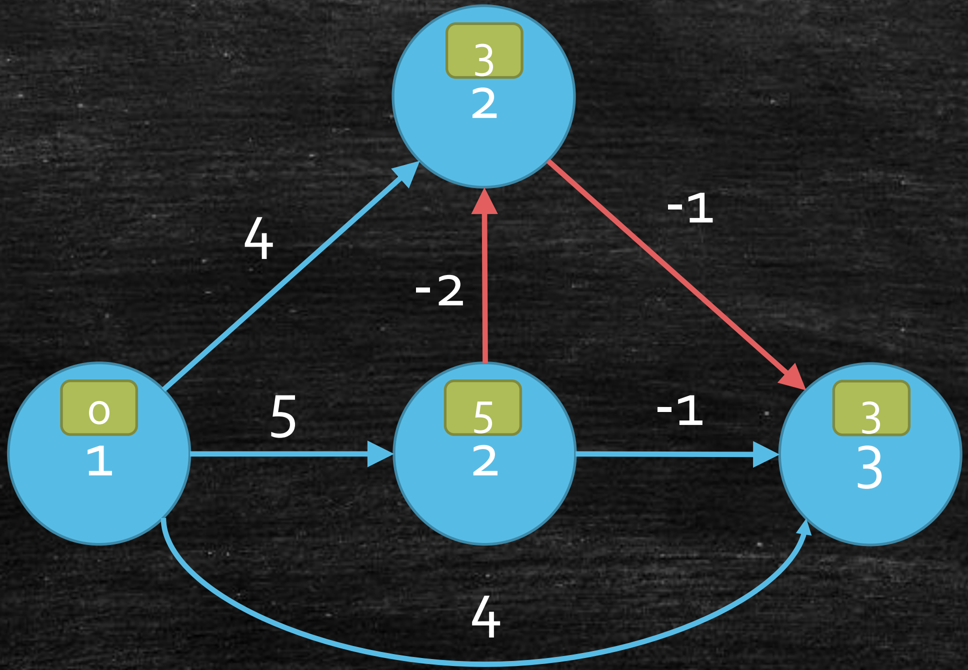
Sample run

Round 1



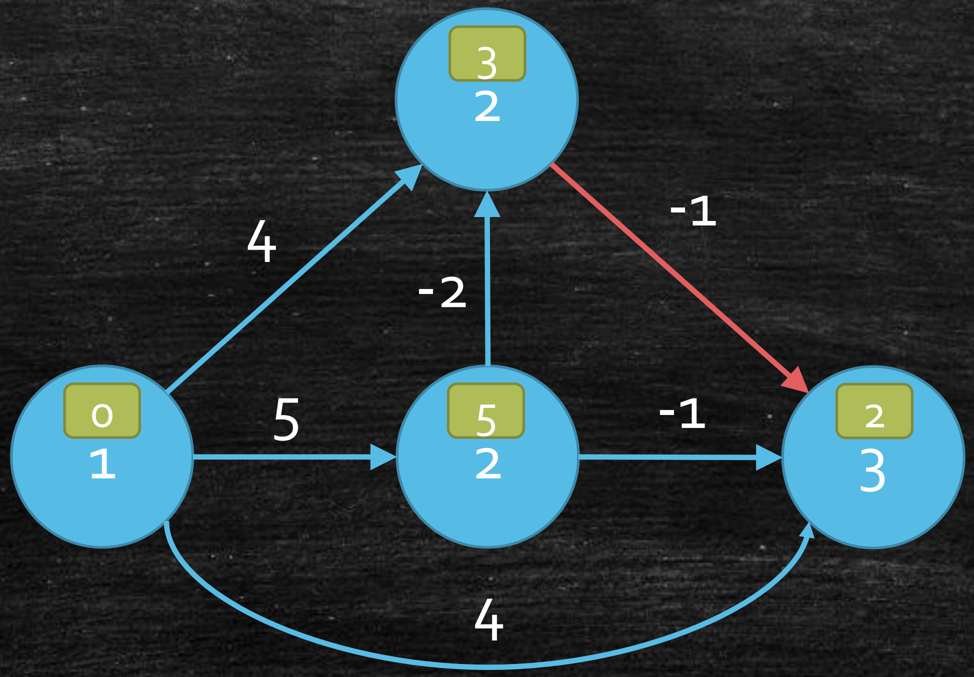
Sample run

Round 2



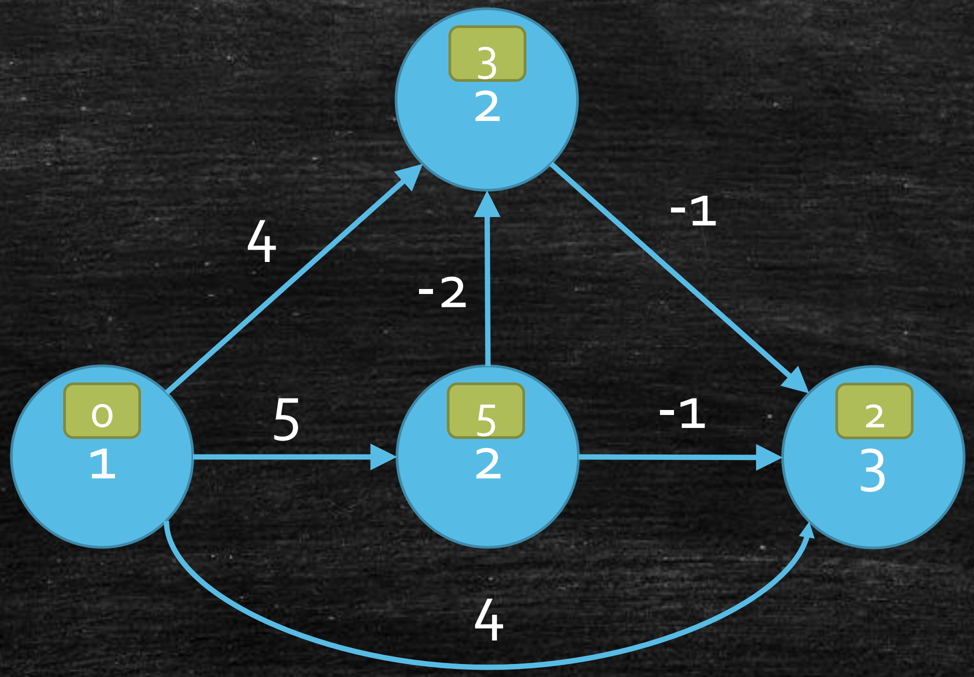
Sample run

Round 3



Sample run

Round 4



Problem

Will it terminate? What is the running time.

Correctness of Bellman-Ford

Lemma 1

After k rounds, $dist(v)$ is the shortest distance of all **k-edge-path**.

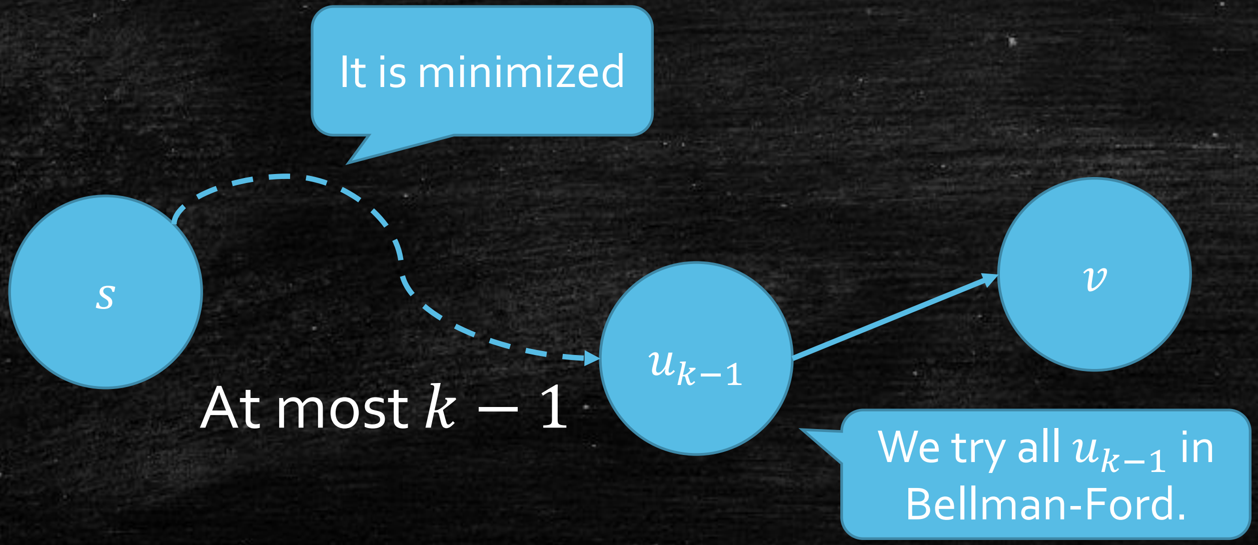
paths with at most k edges.

Proof

- Base case:
  - After 0 rounds, $dist[s]$ is the shortest distance of all **0-edge-path.**
- Induction:
  - Suppose it is true for k-1 rounds.
  - Consider a k-edge-path of v: $(s, u_1, u_2, \dots, u_{k-1}, v)$.

Proof

- Induction:
  - Suppose it is true for k-1 rounds.
  - Consider a k-edge-path of v: $(s, u_1, u_2, \dots, u_{k-1}, v)$.
  - By hypothesis: $dist[u_{k-1}] \leq d(s, u_1, u_2, \dots, u_{k-1})$
  - By Bellman-Ford: $dist[u_k] \leq d(s, u_1, u_2, \dots, u_{k-1}) + d(u_{k-1}, v)$



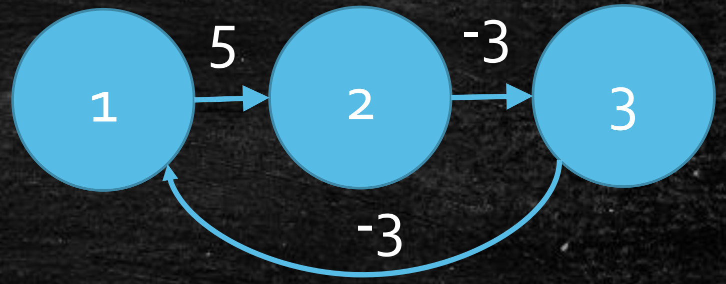
Correctness of Bellman-Ford

Observation 2

The shortest distance of all $|V|$**-edge-path** can not be shorter than the shortest distance of all $(|V|-1)$ **–edge-path** unless there is a **Negative Cycle.**

Proof

- $|V|$-edge-path must contains a cycle
- If the cycle is not negative, go through it do not make the distance smaller.



Negative Cycle

Negative Cycle

- What if G has a negative cycle?
- The shortest distance become **not well defined**!
- The shortest distance can as **small** as we want!

Correctness of Bellman-Ford

Lemma 1

After k rounds, $dist(v)$ is the shortest distance of all **k-edge-path**.

Observation 2

The shortest distance of all $|V|$-**edge-path** can not be shorter than the shortest distance of all $(|V|-1)$ **–edge-path** unless there is a **Negative Cycle.**

Conclusion

After $|V|-1$ rounds, $dist(v)$ is the **shortest distance**, otherwise G has a **Negative Cycle.**

$O(|V| \cdot |E|)$

Refine The Algorithm

Run $|V|$ **rounds** updating, If distance become shorter in the $|V|$-th round, output negative cycle, otherwise, output distance.

Today’s goal

- Learn why **Dijkstra** is wrong when **edge** is negative.
  - Learn to find a **counter example**.
  - Learn to point out the **problems** in the proof.
- Learn **Bellman-Ford**
  - What kind of graphs make it **correct?**
  - Why we only need to consider this kind of graphs?