به‌نام خدا.

$s \in V$ مبدأ

هزینه یال‌های گراف منفی باشد.

single shortest paths

Bellman - Ford.

یال ریلکس

(u, v)

$v.d > u.d + w(u, v)$

$v.d = u.d + w(u, v)$

$v.\pi = u$

predecessor

At most, $\underline{|V| - 1}$ edges in one of our paths.

$|V|$ = # of vertices

$|V|$ or more edges in a path

$\Downarrow$

repeated vertex

$\Downarrow$

cycle

هزینه یالها می‌تواند منفی باشد.   مبدأ $s \in V$

single shortest paths

Bellman-Ford.   بنام خدا.

Initialize-Single-Source(G, s)

for $i = 1$ to $n-1$

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

        Relax(u, v, w)

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

    if $v.d > u.d + w(u, v)$

        return False

return True

یال ریلکس $(\overrightarrow{u, v})$

$v.d > u.d + w(u, v)$

$v.d = u.d + w(u, v)$

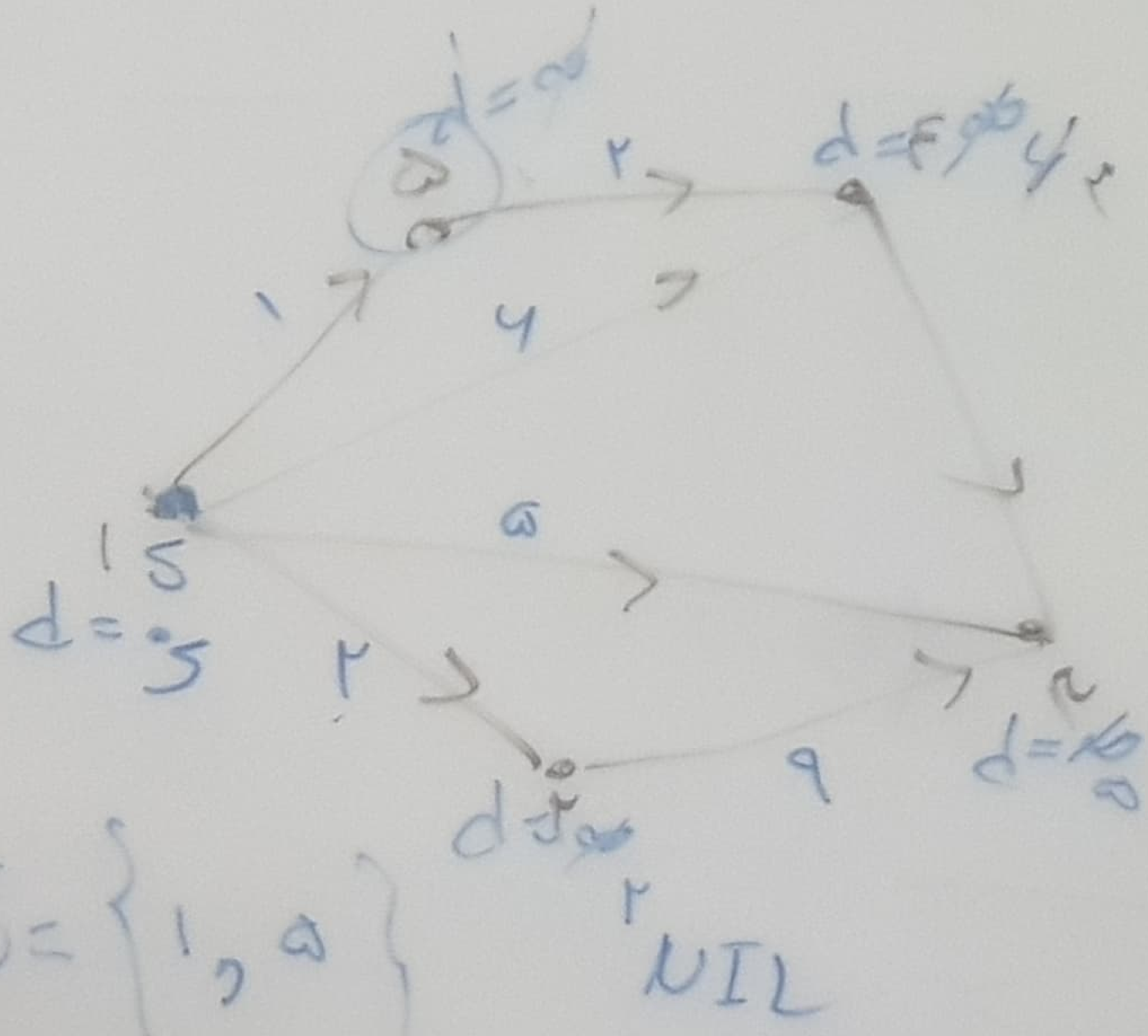
$v.\pi = u$

predecessor

d: مقدار کران بالای برای طول کوتاه‌ترین مسیر از s به v

δ: طول کوتاه‌ترین مسیر

$d = \delta$



$S = \{1, 5\}$

$T = \{\ \}$

بنام خدا.

هزینه یالها می‌توانند منفی باشد. $s \in V$ مبدأ

single shortest paths

Bellman - Ford.

Initialize - Single - Source (G, s)

for $i = 1$ to $n - 1$

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

    Relax (u, v, w)

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

  if $v.d > u.d + w(u, v)$

    return False

return True

یال ریلکس

(u, v)

$v.d > u.d + w(u, v)$

$v.d = u.d + w(u, v)$

$v.\pi = u$

predecessor

(برای تشخیص دور منفی نیز قابل استفاده است)

هزینه یالها و ترانزیشن باشد. میدا $s \in V$

single shortest paths

بنام خدا.

Bellman Ford.

Initialize_Single_Source(G, s)

for $i=1$ to $n-1$

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

Relax(u, v, w)

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

if $v.d > u.d + w(u,v)$

return False

return True

یال ریلکس (u, v)

$v.d > u.d + w(u,v)$

$v.d = u.d + w(u,v)$

$v.\Pi = u$

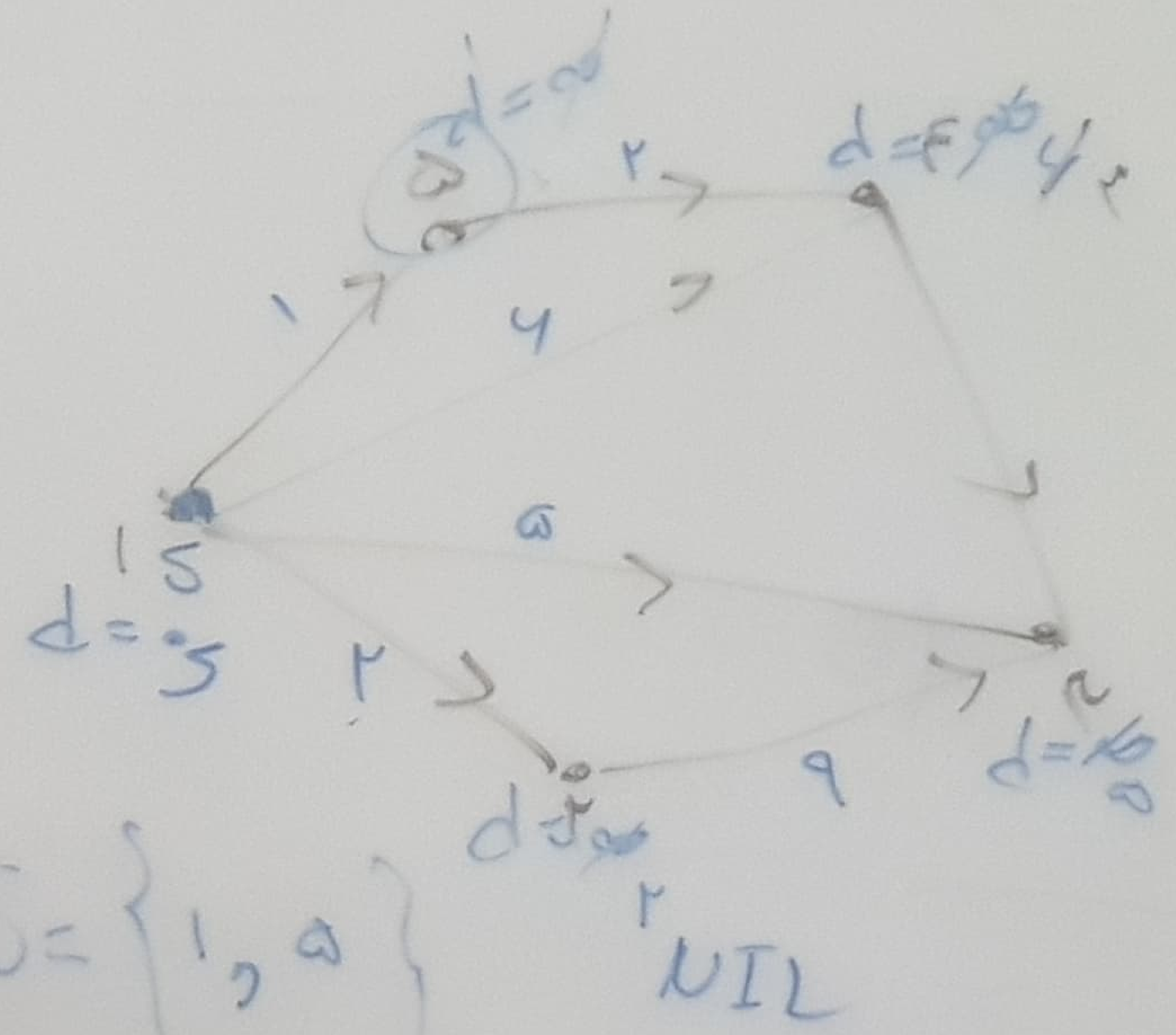
predecessor

یادآوری مثالی از Dij

d: کران بالای برای طول کوتاهترین مسیر از s به v (مقدار)

δ: طول کوتاه ترین مسیر

d = δ



S = {۱, ۵}

T = { }

اثبات:

لم ۱: اگر گراف G دور (با هزینه) منفی که از s قابل دسترس باشد، نداشته باشد، آنگاه الگوریتم بعد از n-1 تکرار جواب بهینه را می‌دهد یعنی

$$\forall v \in V \quad v.d = \delta(s, v)$$

به ازای هر — طول کوتاه‌ترین مسیر از s به v

رأس v را در نظر بگیرید. که $p = \langle v_0, v_1, \dots, v_k \rangle$ یک کوتاه‌ترین مسیر از s به v باشد. $v_k = v$ ، $v_0 = s$

چون p مسیر ساده است پس حداکثر n-1 یال دارد.

یعنی $k \le n-1$. در هر n-1 تکرار حلقه for تمام m یال ریلکس می‌شوند و در میان یال‌های ریلکس شده در تکرار i‌ام حتماً یال (v_{i-1}, v_i) وجود دارد که $i = 1, \dots, k$ و استقرا حکم برقرار است.

Proof We show by induction that after the ith edge of path p is relaxed, we have $v_i.d = \delta(s, v_i)$. For the base case, $i = 0$, and before any edges of p have been relaxed, we have from the initialization that $v_0.d = s.d = 0 = \delta(s, s)$. By the upper-bound property, the value of $s.d$ never changes after initialization.

For the inductive step, assume that $v_{i-1}.d = \delta(s, v_{i-1})$. What happens when edge (v_{i-1}, v_i) is relaxed? By the convergence property, after this relaxation, we have $v_i.d = \delta(s, v_i)$, and this equation is maintained at all times thereafter. ■

درک درست فرآیند:

- در **اولین تکرار**، الگوریتم فقط مسیرهای مستقیم با یک یال از رأس مبدأ s را ریلکس می‌کند.
- در **تکرار دوم**، مسیرهایی با **دو یال** بررسی و به‌روزرسانی می‌شوند. به همین ترتیب، در **تکرار i-ام**، مسیرهایی با حداکثر i یال بهبود می‌یابند.

بنام خدا.

llman ford.

Initialize_Single_Source(G, s)

for i = 1 to n - 1

for each edge(u, v) ∈ E

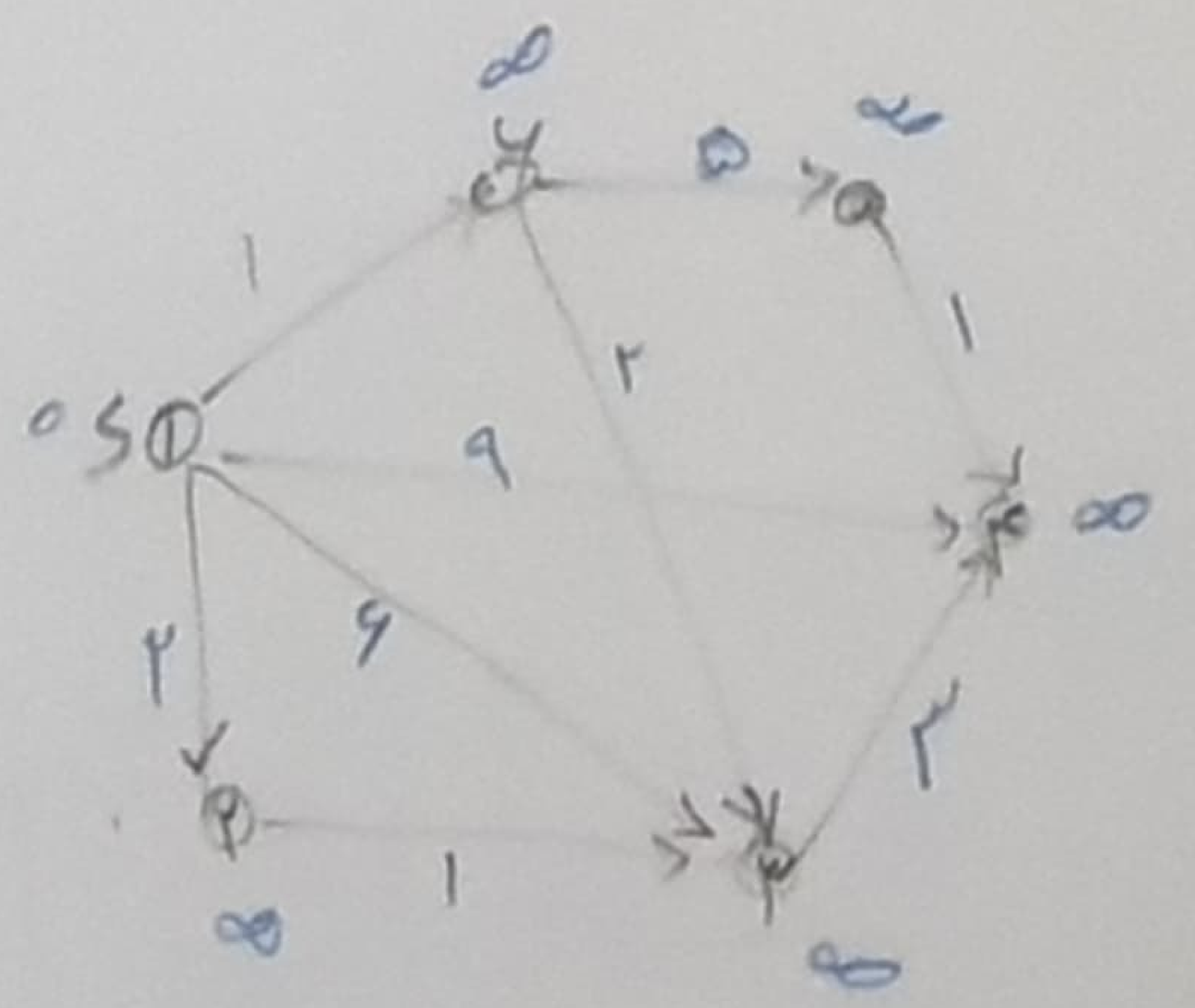
Relax(u, v, w)

for each edge(u, v) ∈ E:

if v.d > u.d + w(u, v)

return False

return True

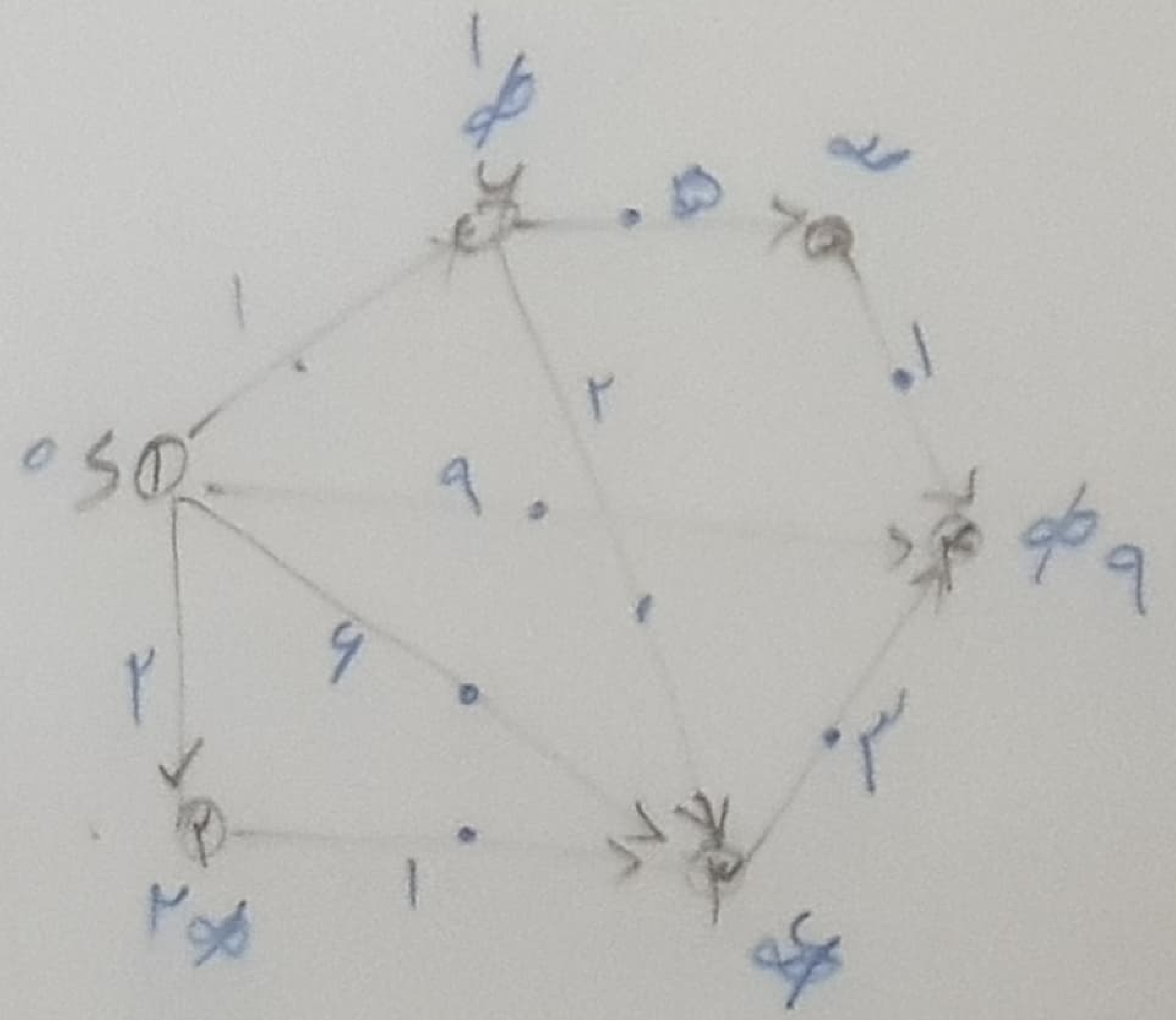


تکرار ۱

بنام خدا

Bellman - Ford.

```
Initialize_Single_Source(G, s)
for i = 1 to n - 1
    for each edge(u, v) ∈ E
        Relax(u, v, w)
for each edge(u, v) ∈ E:
    if v.d > u.d + w(u, v)
        return False
return True
```



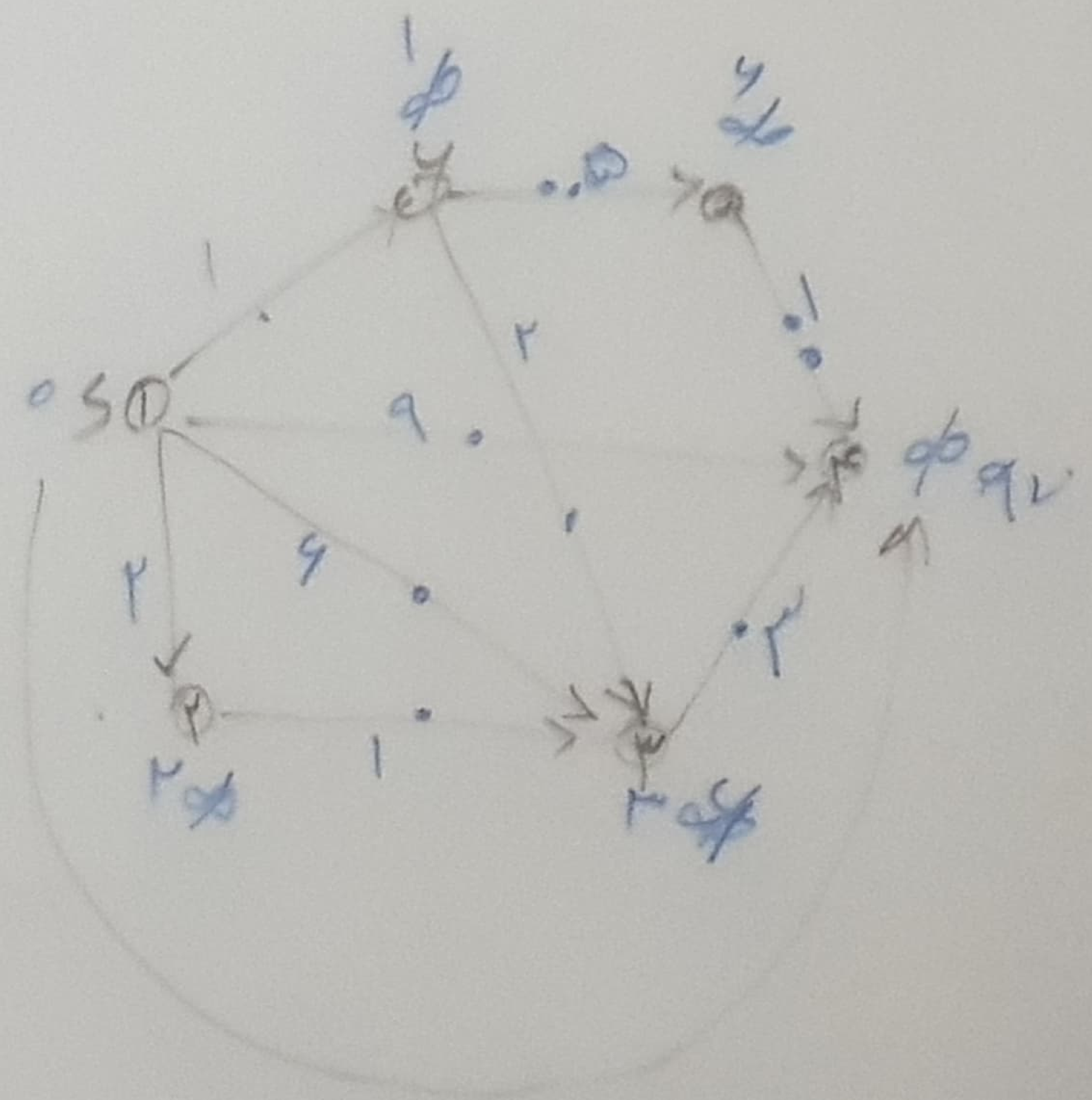
بنام خدا

تکرار ۱

Bellman - Ford.

Initialize_Single_Source(G, s)

```
for i = 1 to n-1
    for each edge (u,v) ∈ E
        Relax(u, v, w)
for each edge (u,v) ∈ E:
    if v.d > u.d + w(u,v)
        return False
return True
```



اثبات:

$$v_i.d \le v_{i-1}.d + w(v_{i-1}, v_i)$$

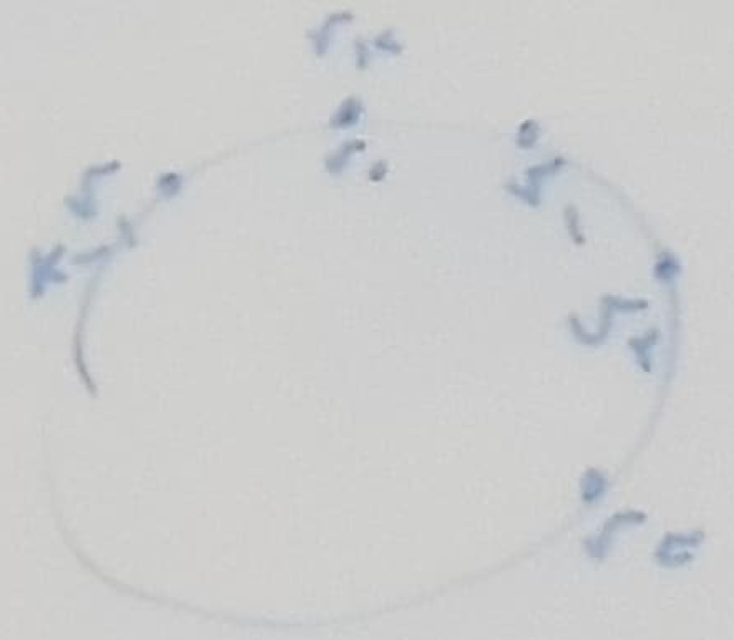
حکم: اگر در گراف G دور منفی با هزینه که از s قابل دسترسی باشد، وجود داشته باشد.

الگوریتم، مقدار False را بر می‌گرداند.

فرض کنیم $C = <v_0, v_1, \dots, v_k>$ دوری باشد که هزینه آن منفی است

$\boxed{v_0 = v_k}$, $\sum_{i=1}^{k} w(v_{i-1}, v_i) < 0$

اثبات از برهان خلف. فرض کنیم الگوریتم True برگردانده باشد یعنی برای همه یال‌ها



(u,v)

$$v.d \le u.d + w(u,v)$$

برای تمام رئوس روی دور C این نامساوی را جمع می‌زنیم

$$\sum_{i=1}^{k} v_i.d \le \sum_{i=1}^{k} v_{i-1}.d + w(v_{i-1}, v_i)$$

$$\sum_{i=1}^{k} v_i.d \le \sum_{i=1}^{k} v_{i-1}.d + \sum_{i=1}^{k} w(v_{i-1}, v_i)$$

ولی $\sum_{i=1}^{k} v_i.d = \sum_{i=1}^{k} v_{i-1}.d$ یعنی

$$\sum_{i=1}^{k} w(v_{i-1}, v_i) \ge 0$$

که تناقض است