$O \quad m \rightsquigarrow n^5$

$G = (V, E)$

s, t

$c > 0$

اگر مقادیر ظرفیت ها عدد صحیح باشد، هر بار

ارسال جریان حداقل یک واحد مقدار کل جریان

افزایش پیدا می کند.

شبه چندجمله ای

$f = 0 \rightsquigarrow G_f \longrightarrow s, t$

مسیر افزایشی P

$\longrightarrow f \uparrow f_P$

$c(P) = \min\{c_f(u,v), (u,v) \in P\}$

$O(n^{\text{چندجمله ای برحسب } n})$

n تعداد رأس ها

m تعداد یال ها

$O($ تعداد اعمال برحسب اندازه ورودی $)$

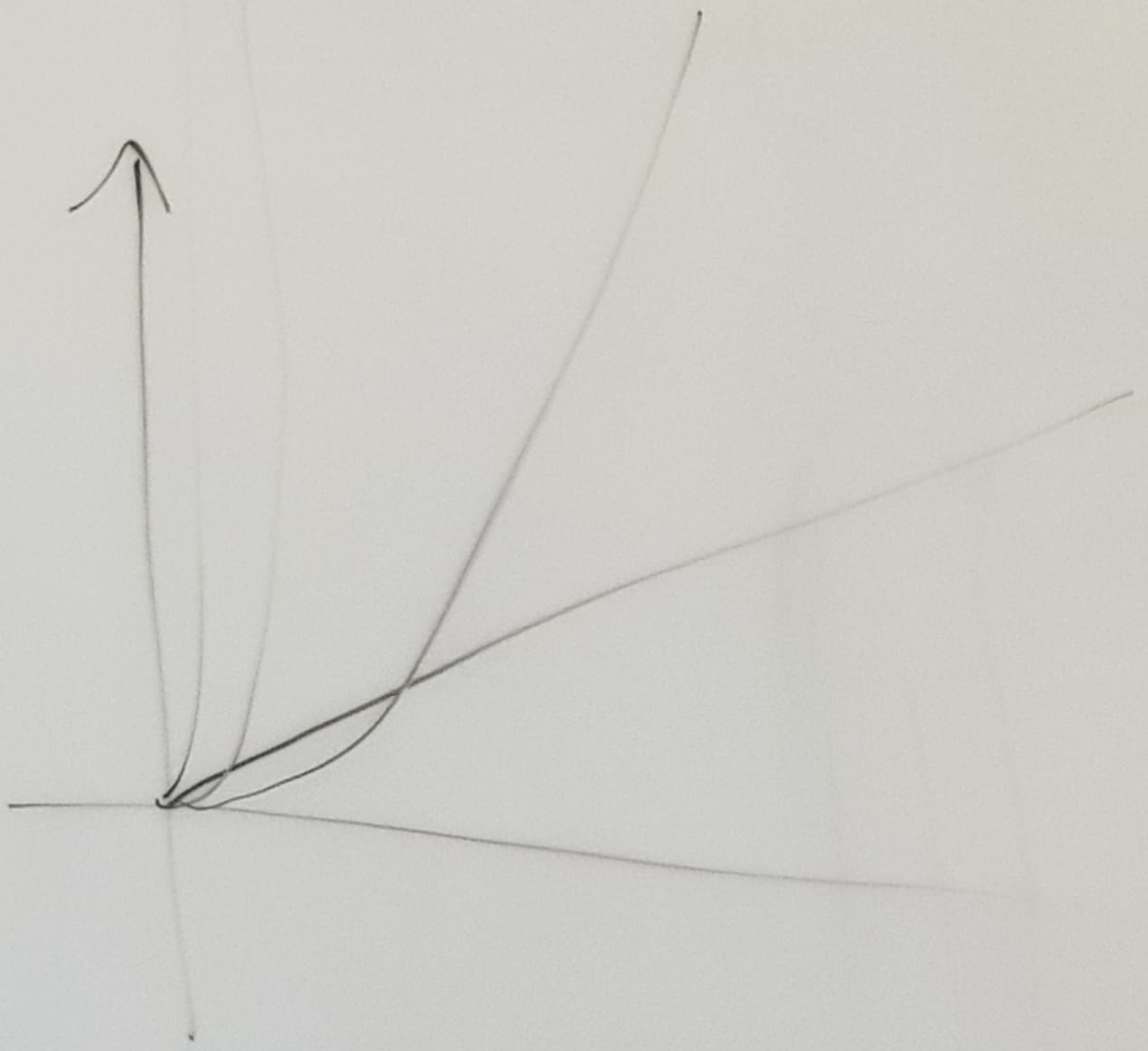
$\log n$

n

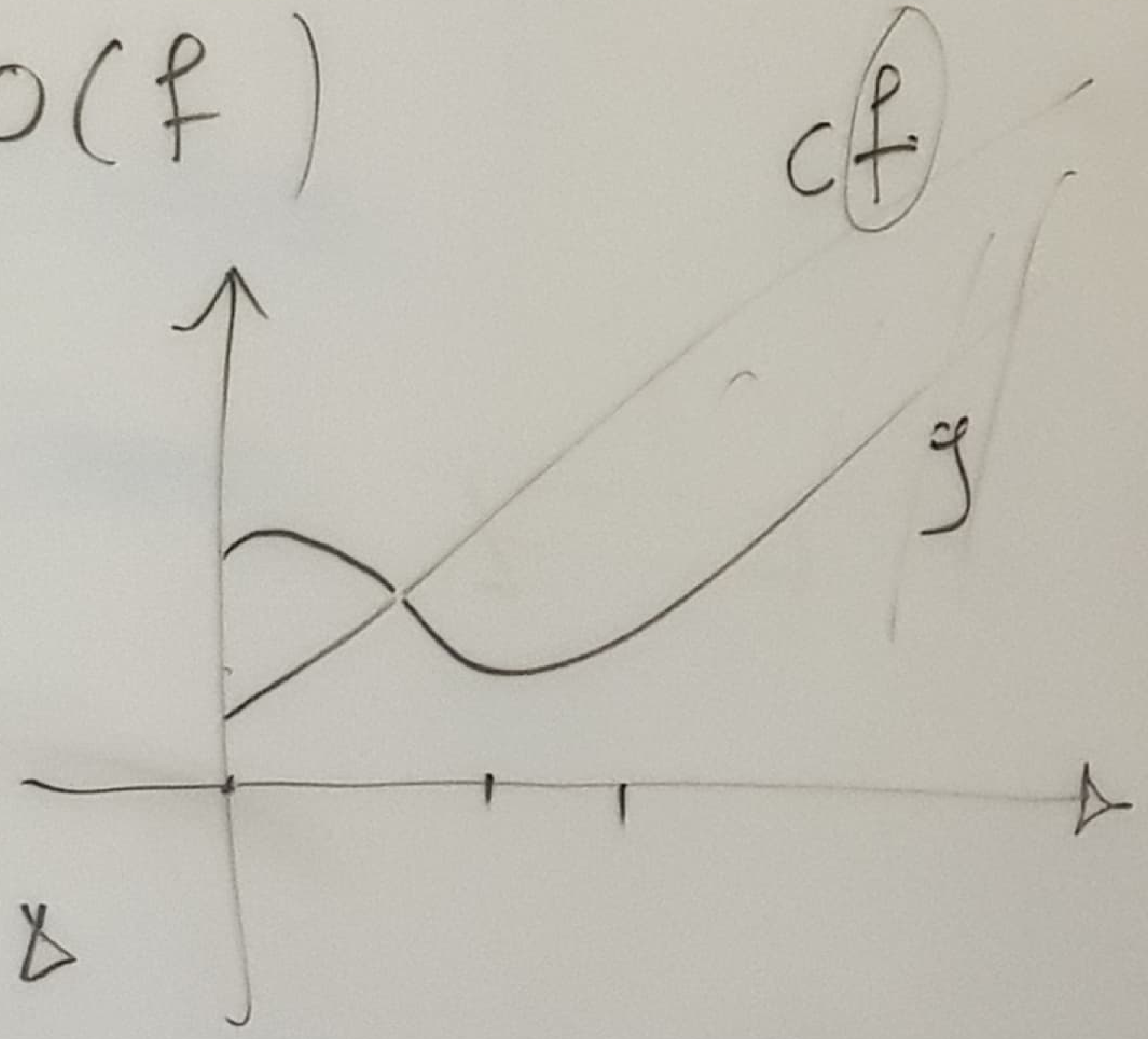
n^2

2^n

$n!$



$g \in O(f)$



$O\ m \rightsquigarrow n^2$

$G = (V, E)$

s , t

$c \geq 0$

$f = 0 \rightsquigarrow G_f \longrightarrow s , t$ مسیر افزایشی p BFS $\longrightarrow$ طول هر یال یک (مسیر با کمترین تعداد یال)

f ? مسیر با بیشترین ظرفیت ؟

$c(p) = \min\{c_f(u,v) , (u,v) \in p\}$

$f =$

اگر مقادیر ظرفیت ها عدد صحیح باشد، هر بار

ارسال جریان حداقل یک واحد مقدار کل جریان

افزایش پیدا می کند.

$|f^*|$

الگوریتم ادموندز کارپ Edmonds Karp

$O(nm^2)$

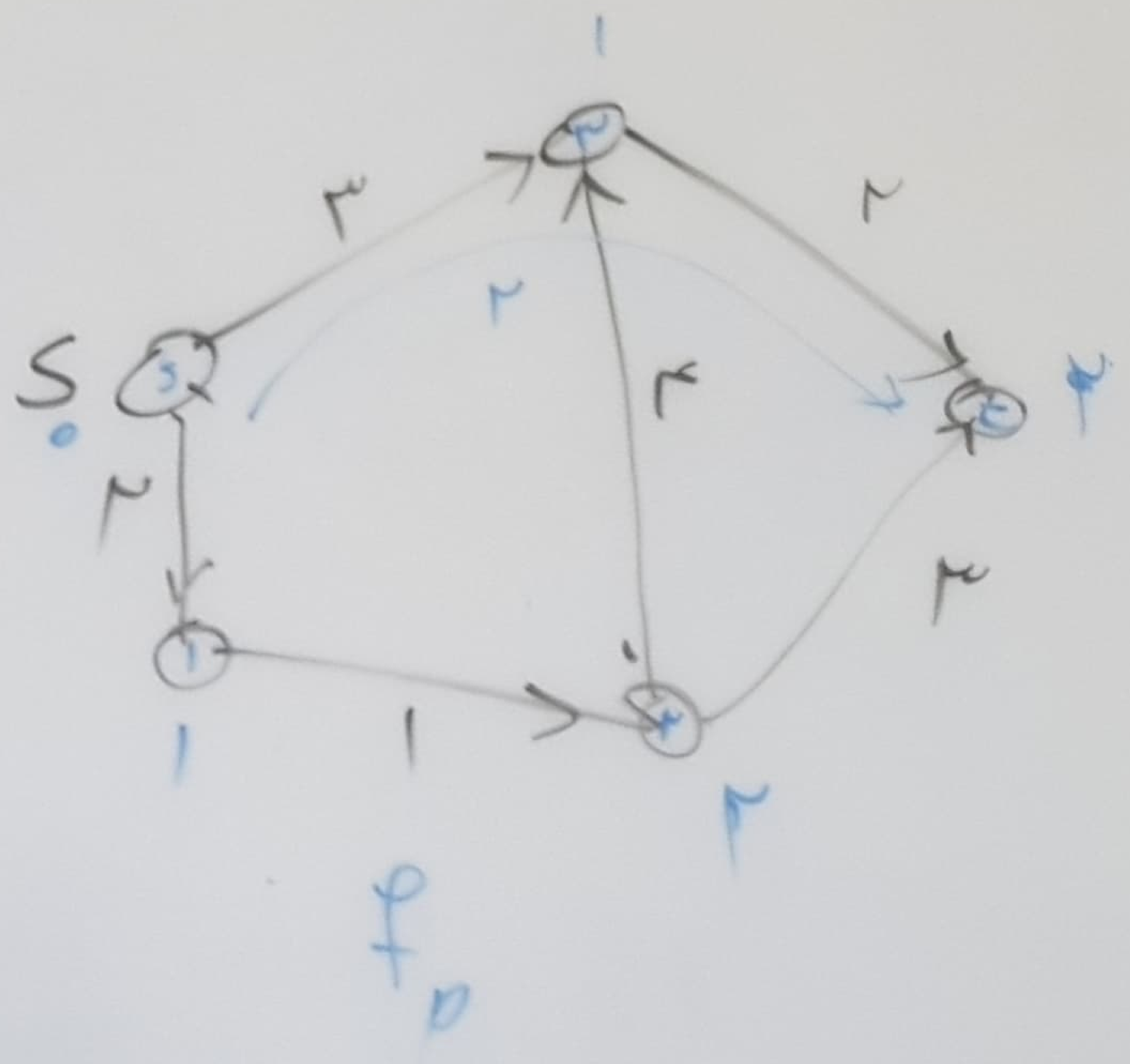
الگوریتم Edmons-karp

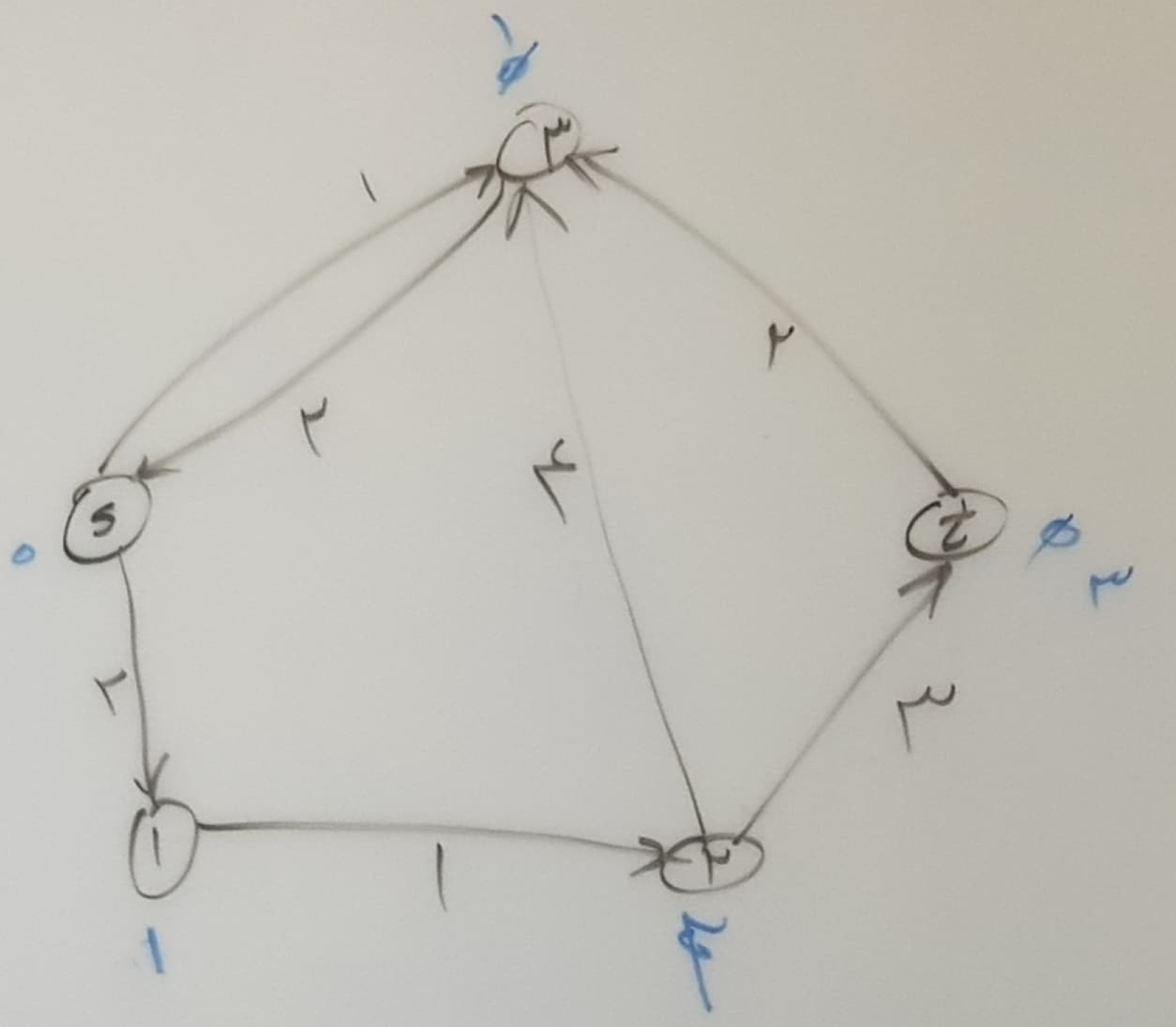
همان الگوریتم فورد فالکرسون است که برای پیدا کردن مسیر افزایشی در شبکه باقی مانده از BFS استفاده می‌کند.

طول کوتاه‌ترین مسیر از u به v = $\delta(u,v)$

$\delta(s,u)=0$

لم: با اجرای الگوریتم ادموندز کارپ روی شبکه $G=(V,E)$ با رأس مبدأ s و رأس مقصد t، برچسب‌های $\delta_f(s,v)$ (طول کوتاه‌ترین مسیر از s به v در شبکه باقی مانده G_f) با افزایش جریان به طور صعودی است.





لم: عمق BFS رئوس بعد از هر اجرا بیشتر یا مساوی می‌شود

الگوریتم Edmons-karp

همان الگوریتم فورد فالکرسون است که برای پیدا کردن مسیر افزایشی در شبکه باقی مانده از BFS استفاده می کند.

$\delta(u,v) =$ طول کوتاه‌ترین مسیر از u به v

$\delta(s,u)=0$

لم: با اجرای الگوریتم ادموندز کارپ روی شبکه $G=(V,E)$ با رأس مبدا s و رأس مقصد t، برچسب های $\delta_f(s,v)$ (طول کوتاه‌ترین مسیر از s به v در شبکه باقی مانده G_f) با افزایش جریان به طور صعودی است.

اثبات $\quad f \longrightarrow G_f \longrightarrow f'$

گیر رأس v که $\delta_{f'}(s,v) < \delta_f(s,v)$

و از میان تمام رئوس این چنینی، v رأس با کمترین برچسب باشد.

$p: s \rightsquigarrow u \to v$

کوتاه‌ترین مسیر از s به v در $G_{f'}$ باشد.

$$(u,v) \in E_{f'}$$

$$\delta_{f'}(s,u) = \delta_{f'}(s,v) - 1$$

پس u راسی نیست که به موجب آن کاهش پیدا کرده باشد و داریم

$$\delta_{f'}(s,u) \geq \delta_f(s,u)$$

ادعا می‌کنیم $(u,v) \notin E_f$ زیرا در غیر این صورت

$$\delta_f(s,v) \le \delta_f(s,u) + 1$$

$$\le \delta_{f'}(s,u) + 1$$

$$= \delta_{f'}(s,v)$$

که با فرض ما تناقض دارد.

پس یعنی در G_f از v به u جریان ارسال شده و به عبارت دیگر یال $(v,u) \in E_f$ بوده است.

یعنی یال (v,u) از G_f در کوتاه‌ترین مسیر از s به u قرار گرفته است.

و با توجه به اینکه هر زیرمسیر، مسیر بهینه، بهینه است داریم

$$\delta_f(s,v) = \delta_f(s,u) - 1$$

$$\le \delta_{f'}(s,u) - 1$$

$$= \delta_{f'}(s,v)$$

که تناقض است و راسی مانند v وجود ندارد که به موجب آن کاهش پیدا کند.

الگوریتم Edmons-karp

قضیه: تعداد کل مسیرهای افزایشی در اجرای الگوریتم ادموندز-کارپ $O(nm)$ است.

یال (u,v) در شبکه باقی‌مانده G_f را بحرانی می‌گوییم اگر
تعریف می‌کنیم.
برای مسیر افزایشی p: $c_f(p)=c_f(u,v)$

پس با ارسال جریان روی p، یال (u,v) حذف می‌شود.

* دقت کنید در هر بار ارسال جریان حداقل یک یال بحرانی می‌شود.

* نشان می‌دهیم هر یال حداکثر $\frac{n}{2}$ بار می‌تواند بحرانی شود.

فرض کنید یال $(u,v)\in E$ باشد. از آنجایی که
مسیرهای افزایشی، کوتاه‌ترین مسیرها هستند، اولین
باری که یال (u,v) بحرانی شود داریم

$$\delta_f(s,v)=\delta_f(s,u)+1$$

با ارسال جریان یال (u,v) حذف می‌شود.
و تا زمانی که جریان از v به u برنگردد مجدداً یال
(u,v) ظاهر نمی‌شود.

اگر f' جریانی باشد که در شبکه باقی‌مانده آن یال (v,u)
در مسیر افزایشی قرار گرفته باشد خواهیم داشت

یعنی یال (v,u) از کوتاه‌ترین مسیر است و داریم

$$\delta_{f'}(s,u) = \delta_{f'}(s,v) + 1$$

و از آنجایی که $\delta_f(s,v) \le \delta_{f'}(s,v)$ داریم

$$\delta_{f'}(s,u) \geq \delta_f(s,v) + 1$$

$$= \delta_f(s,u) + 2$$

پس از آنجایی که (u,v) بحرانی شود تا بار بعدی که بحرانی شود برچسب u حداقل ۲ واحد زیاد می‌شود.

از آنجایی که $u \neq t$ طول کوتاه‌ترین مسیر از s به u حداکثر $n-2$ است یعنی حداکثر $\frac{n-2}{2}$ بار یال (u,v) می‌تواند بحرانی شود با احتساب اولین مرتبه بحرانی شدن $\frac{n}{2}-1+1$ یا همان $\frac{n}{2}$.

پس تعداد کل مسیرهای افزایشی از مرتبه $O(nm)$ است.

اجرای BFS، از مرتبه $O(m)$ است که نتیجه می‌دهد زمان اجرای کل الگوریتم

$$O(nm^2)$$

خواهد بود.