

resource vector =  $\vec{R}$  = جمع کل منابع

سوال 1:

$\Rightarrow$

	A	B	C	D
	7	12	9	9

این کار را می توانیم ترتیب را برای ابرای بررسی  
ما پیدا کنیم که منجر به دو حالت غیر امکان پذیر  
و تمام بررسی ها با موفقیت ابرای شوند پس حالت امن است.

current:

	A	B	C	D
P <sub>1</sub>	1	2	3	1
P <sub>2</sub>	1	1	1	3
P <sub>3</sub>	1	1	1	1
P <sub>4</sub>	1	1	1	1
P <sub>5</sub>	3	2	2	1

$\Rightarrow \sum \text{ منابع} = \text{بردار} = (6, 8, 7, 8)$   
allocated

$\Rightarrow \text{all resources} - \text{allocated} = \text{available} = (7, 12, 9, 9) - (6, 8, 7, 8)$   
 $= (1, 4, 1, 1)$

Need = max Requests - current assigned =

	A	B	C	D
P <sub>1</sub>	1	5	2	4
P <sub>2</sub>	5	7	6	6
P <sub>3</sub>	1	1	1	1
P <sub>4</sub>	2	4	1	3
P <sub>5</sub>	2	5	5	5

Need =

$\Rightarrow \text{max request} =$

$\Rightarrow \text{available} = (1, 4, 1, 1) + \text{current assigned } [P_3] = (2, 5, 1, 3)$

$\text{new available} = (2, 5, 1, 3) + (0, 2, 1, 1) = (2, 7, 2, 4)$  P<sub>4</sub> ②

$\text{available} = (2, 7, 2, 4) + (1, 2, 3, 1) = (3, 9, 5, 5)$  P<sub>1</sub> ③

$\text{available} = (3, 9, 5, 5) + (\text{current}(P_5)) = (6, 11, 7, 6)$  P<sub>5</sub> ④

⑤ P<sub>2</sub>، النظام رسیم؛ باید در آخرین مرحله برابر available به دست آمده با بردار تمام  
منابع به عدد اولی برابر شود تا بتوانیم حالت امن است و به دو حالت برنخوریم:

آزیدار بود و در جمع کل منابع  $\text{current}(P_2) = (7, 12, 9, 9) = (6, 11, 7, 6) +$



ج) بررسی می‌کنیم: یک منبع از B را به P<sub>1</sub> اختصاص می‌دهیم:

	A	B	C	D
P <sub>1</sub>	1	3	3	1
P <sub>2</sub>	1	1	1	3
P <sub>3</sub>	1	1	0	2
P <sub>4</sub>	0	2	1	1
P <sub>5</sub>	3	2	2	1

available = (1, 3, 1, 1) برابر

نتایج تخصیص باقی new

=> need matrix =

	A	B	C	D
P <sub>1</sub>	2	4	2	4
P <sub>2</sub>	5	7	6	6
P <sub>3</sub>	1	2	1	1
P <sub>4</sub>	2	4	1	3
P <sub>5</sub>	2	5	5	5

① <= P<sub>3</sub>, اعلام می‌دهیم ✓

=> av = (2, 4, 1, 3)

② <= P<sub>4</sub>, اعلام می‌دهیم ✓

=> av = (2, 6, 2, 4)

av = (3, 9, 5, 5) ✓ P<sub>1</sub> ③ <=

=> الان آخر: بدلیل اینکه توانستیم P<sub>1</sub> را در مرحله سوم هم اعلام بدهیم و به برابر available مثل

قسمت ان رسیدیم، می‌توانیم بگوییم: پذیرش این خواسته بدلیل عدم ایثار هیچ‌گونه تبعیضی فاقد مشکل بوده

و می‌توان با درخواست فوق موافقت کرد.

پس برای بررسی درخواست را اجابت می‌کنیم:

ماتریس Need جدید:

	A	B	C	D
P <sub>1</sub>	2	5	2	4
P <sub>2</sub>	5	7	6	6
P <sub>3</sub>	1	2	1	1
P <sub>4</sub>	2	4	1	3
P <sub>5</sub>	2	3	5	5

=> برابر available = (1, 2, 1, 1)

=> available = (2, 3, 1, 3): P<sub>3</sub> ② <=

allocated matrix = current ✓

	A	B	C	D
P <sub>1</sub>	1	2	3	1
P <sub>2</sub>	1	1	1	3
P <sub>3</sub>	1	1	0	2
P <sub>4</sub>	0	2	1	1
P <sub>5</sub>	3	4	2	1

② <=: هیچ‌یک از سایر پروسه‌ها را نمی‌توانیم

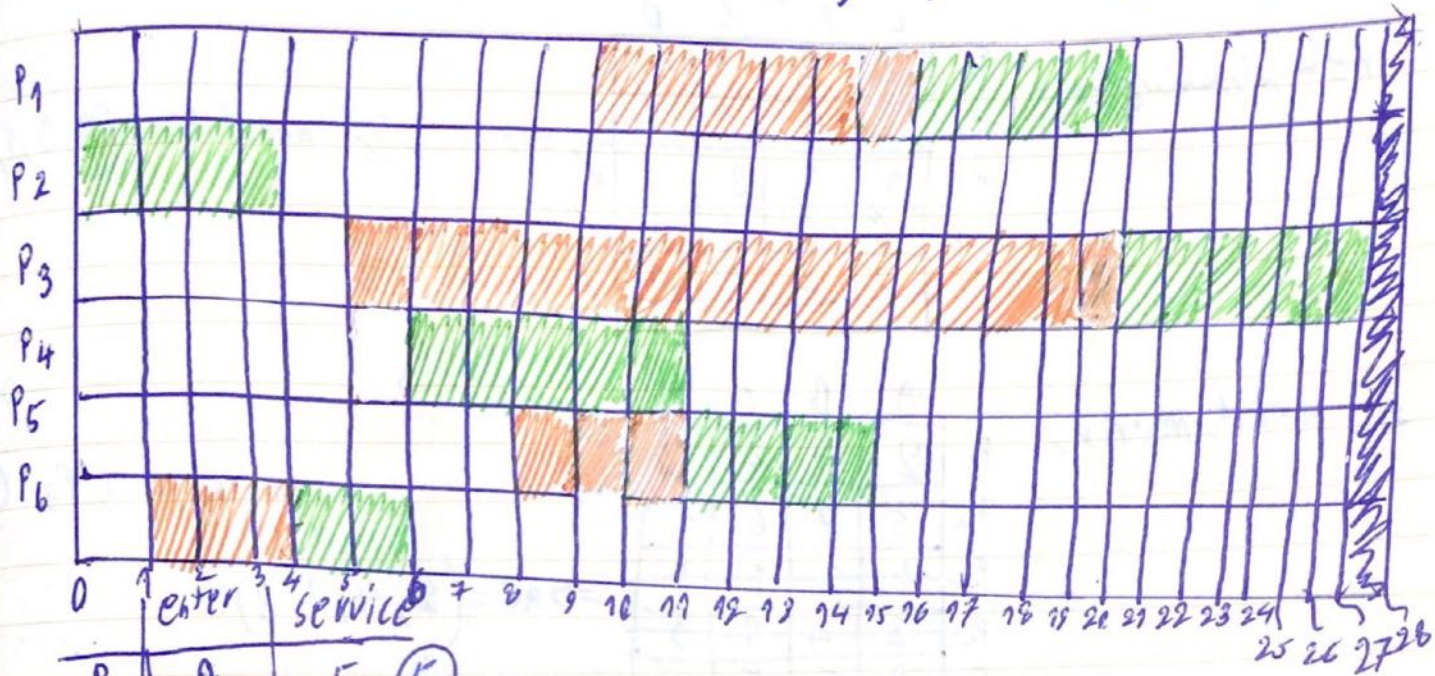
اجرا کنیم به هیچ روش میسر نیست؛ به دلایل برقرار داریم

پس این حالت (اختصاص دومین B افاده به P<sub>5</sub>) حالت ناممکن است و منجر به در لاک تراشه شد.



exclusive approach

SJF is Non-Preemptible scheduling Algorithm (نفا) (2) سرال



P1	9	5	(5)
P2	0	4	(1) ✓
P3	5	7	(3)
P4	6	5	(4)
P5	8	4	(4)
P6	1	2	(2) ✓

average waiting Time

$$= \frac{6 + 15 + 3 + 3}{6} = \frac{27}{6} = \frac{9}{2} = 4.5$$

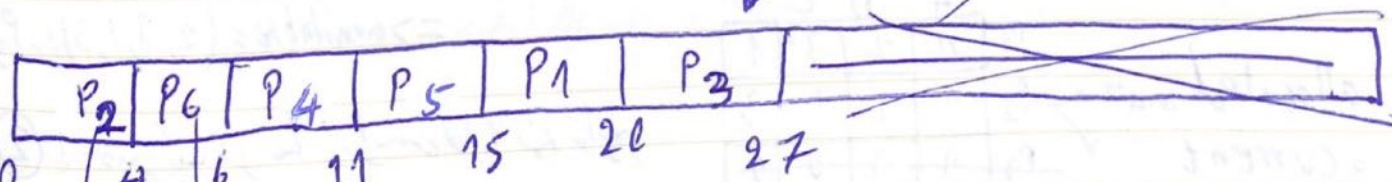
average Turn-around Time

$$= \frac{4 + 5 + 7 + 5 + 22 + 11}{6} = \frac{64}{6} = 10.67$$

زمان ورود - زمان ابر - زمان خروج  
(P) = 6

ردت بستر ← از روی

زمان ورود - زمان خروج  
P2 = 6  
نکات چارت ساده



$$= (4 - 4 - 0) + (6 - 2 - 1) + (11 - 5 - 6) + (15 - 4 - 8) + (20 - 5 - 9) + (27 - 7 - 5)$$

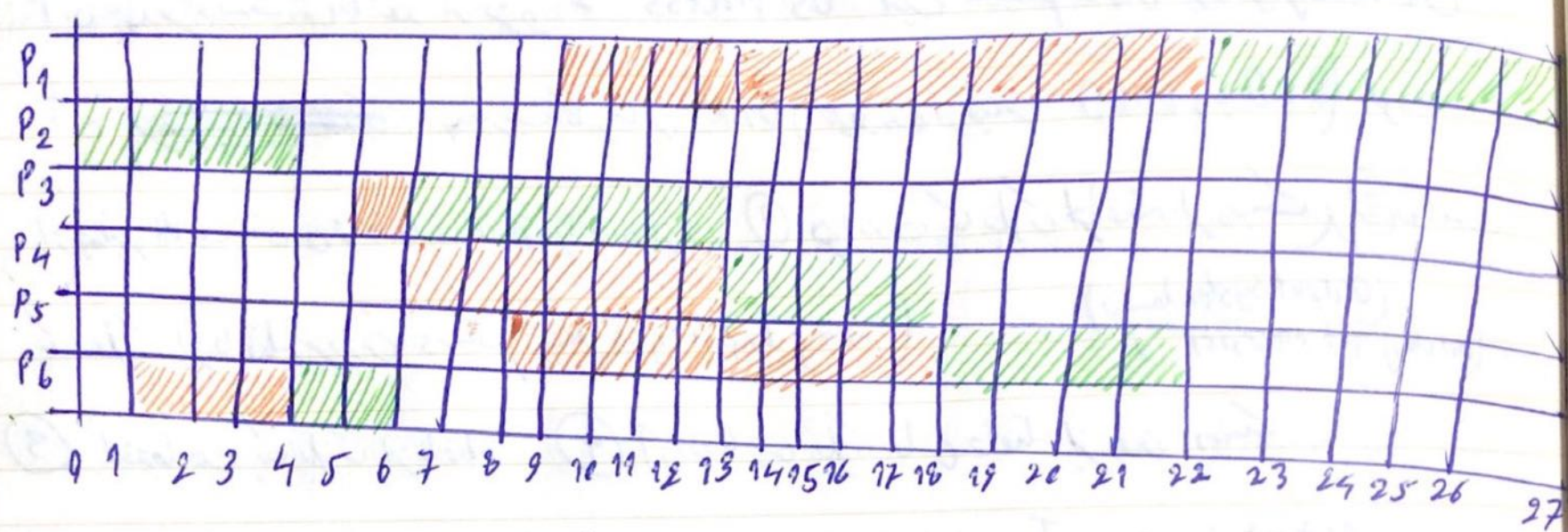
6

$$= 0 + 3 + 0 + 3 + 6 + 15 = 27$$

$$\sqrt{4.5} = \frac{27}{6} = \frac{9}{2} = 4.5$$



یا حیار اولویت این روش مناسب کسر  $\frac{5+13}{5}$  هنگام ورود یک پراسس جدید و تا اتمام پراسس فعلی است:



$0s: \rightarrow P_2 \quad \frac{4+0}{4} = 1 \checkmark \quad \left\{ \begin{array}{l} 4s: \text{ورود } P_6 \text{ در } 1 \quad P_2 \text{ اتمام: } (P_6) \frac{2+3}{2} = 2,5 \end{array} \right.$   
 $6s: \quad P_6 \text{ اتمام} \quad 5s \text{ ورود } P_3 \text{ در } 5 \quad 6s \text{ ورود } P_4 \text{ در } 6 \Rightarrow (P_3) \frac{7+1}{7} = 8/7 \quad P_4: \frac{5+0}{5} = 1$   
 $13s: \quad P_3 \text{ اتمام} \quad \text{ورود تا} \Rightarrow P_1 = \frac{5+4}{5} = 1,8 \quad (P_4) \frac{5+7}{5} = 12/5 = 2,4 \quad P_5: \frac{4+5}{4} = 2,2$   
 $16s: \quad P_4 \text{ اتمام} \Rightarrow P_1 = \frac{5+9}{5} = 2,8 \quad (P_5) \frac{4+10}{4} = 3,5 \quad \left\{ \begin{array}{l} (P_1) \checkmark \end{array} \right.$

$P_2 \rightarrow P_6 \rightarrow P_3 \rightarrow P_4 \rightarrow P_5 \rightarrow P_1$

زمان انتظارها:  $\frac{\text{جمع تارقبها}}{6} = \frac{\sum (\text{زمان ورود} - \text{زمان اجرا})}{6} = \frac{3+1+7+10+13}{6} = \frac{34}{6}$

زمان گردش:  $\frac{\text{سبزهها} + \text{جمع تارقبها}}{6} = \frac{34 + \text{service time}}{6} = \frac{34 + 27}{6} = \frac{61}{6} = 10,1666$

HRRN زمان حاس انتظار و گردش کار بیشترین نسبت به SJF (بیشترین) دارد.



(بافت اول)

سوال سوم: در سیستم عامل نسل سوم *online scheduling* یک از بولده های اصلی

آن مدیر دیسک می باشد که هنوز 7 *OS process* این سیستم عامل بود و وظیفه اصلی

آن ~~مدیر دیسک~~ مدیریت کارهای *Disk* (مجموعه از *HD* و *VD*) بوده

از جمله وظایف آن می تواند موارد فوق را نام برد: ① خواندن یک بلوک حافظه از دیسک ② نوشتن

یک بلوک از حافظه به روی دیسک (توسط *Input system* و *input*) و (توسط *Output system* و *job processor*)

③ اختصاص یک بلوک از حافظه ④ آزادسازی بلوک های حافظه به روی دیسک

*Disk Manager* به طور مستقیم با 3 *OS process* *Input system*، *Output system*،

*job processor* در ارتباط بوده و به طور غیر مستقیم با استفاده از روال های *wait* و *free* با

*Coordinator* که یک از پراسس های سیستم عامل است در ارتباط است و دیسک منیجر

دراک های صف به نام *Disk transfer Queue* بوده که در آن درخواست های انتقال در یک

به پروسس گفته شده را در آن نگهداری میکند و هنگامی که *cpu* به آن نگاه میدارد می کند

تا به درخواست های تمام *entry* های صف رسیدگی کند و به نوبت هر یک را *free* میکند و در نهایت

که هیچ درخواست دیگری در صف باقی نمانده، خودش را *wait* (بلوک) میکند تا *cpu* بکار نشود

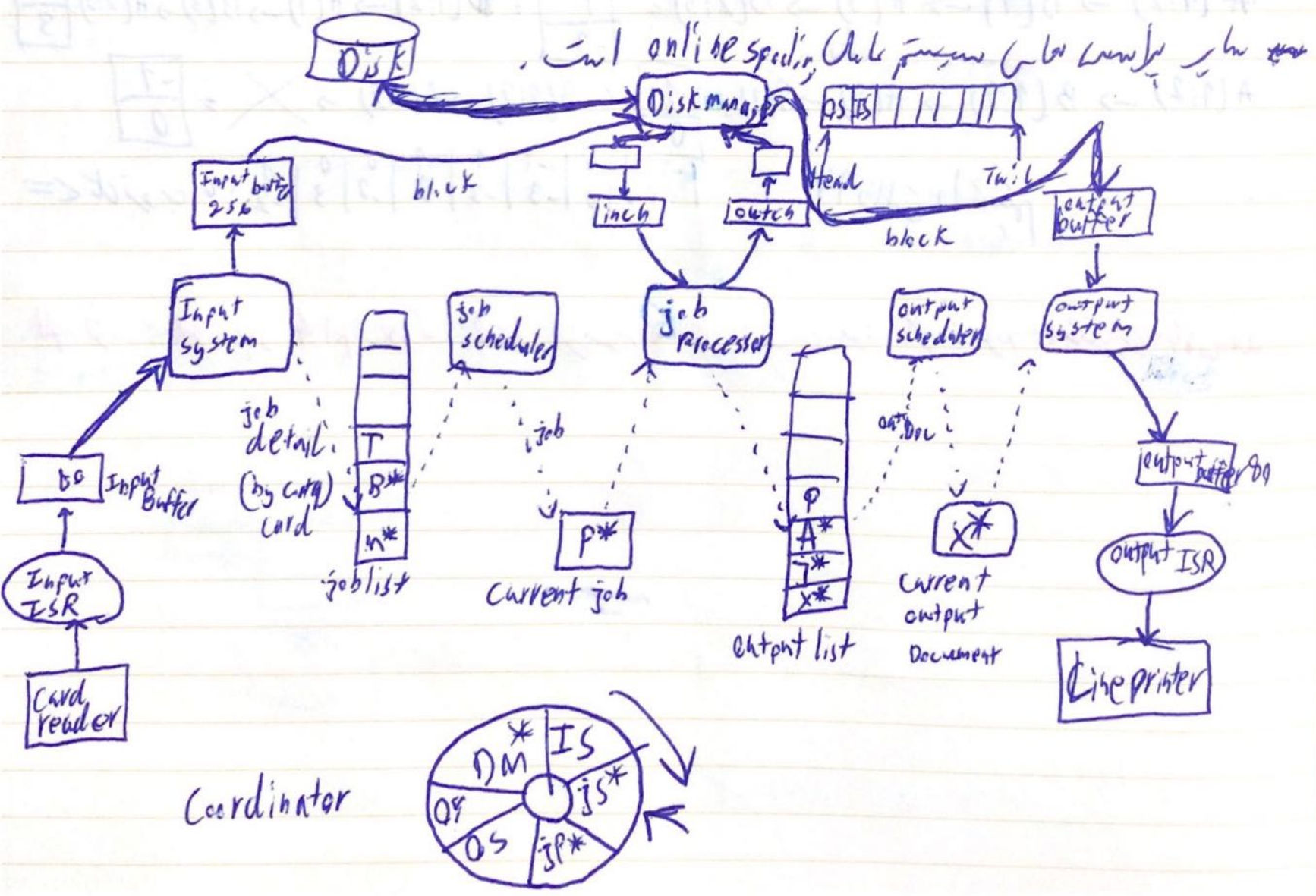
و به سایر پراسس های که به نوبتشان نیاز است توسط *Coordinator* پاسکاری بشود.

ورودی های دیسک منیجر به صورت ~~بلوک~~ بلوک های است که از اینفرم های به اندازه بلوک های حافظه ~~شست~~

می گیرد و آید و داده که در دیسک با اندازه باقی مانده بلوک های حافظه پاک می باشد.



ایستاد (موم): CPU توسط OS process این به نام Coordinator (هماهنگ کننده) که با تمام دیگر  
 6 پرایس سیستم عامل on-line در ارتباط است (به صورت غیر مستقیم) هر پرایس که در صورت نیاز  
 به دیگر پرایس های سیستم عامل احتیاج داشته باشد این نیاز خود را به وسیله روال free به  
 هماهنگ کننده اطلاع می دهد و او آن پروسس مدنظر را در لیست پرنش خود شماره داری کند  
 و پس از اینکه پرایس جاری کارت به اتمام می رسد، از آن بیت که پرایس های سیستم عامل در حال  
 همکاری با هم هستند (بر خلاف پرایس های جزئی) برای اینکه بتواند CPU را متوقف و بیکار کند  
 ندارد قدرت را با یک روال wait بلا می کند تا Coordinator و CPU را بیت دیگر پرایس ها  
 که به نوبت در سیستم احتیاج است بچند و به آن اشاره بکند این نوع روال کردن هم این





$$i=0;$$

A	B
1 $i++;$	1 $i--;$
2 $\text{print}(i);$	2 $\text{print}(i);$
3 $i-=2;$	3 $i+=3;$

$$A \rightarrow B : \begin{bmatrix} 1 \\ -2 \end{bmatrix}$$

سوال 4

↑ A برای سطر اول  
↑ B کل

$$A(1) \rightarrow B \rightarrow A(2:3) = \begin{bmatrix} 0 \\ 3 \end{bmatrix}$$

↓ سطرهای 2 و 3

$$B \rightarrow A : \begin{bmatrix} -1 \\ 3 \end{bmatrix}$$

$$B(1) \rightarrow A \rightarrow B(2:3) = \begin{bmatrix} 0 \\ -2 \end{bmatrix}$$

$$A(1:2) \rightarrow B \rightarrow A(3) = \begin{bmatrix} 1 \\ -2 \end{bmatrix} \Rightarrow \text{دو سطر اول A، کل B، سطر 3}$$

$$B(1:2) \rightarrow A \rightarrow B(3) = \begin{bmatrix} -1 \\ 3 \end{bmatrix}$$

$$A(1) \rightarrow B(1) \rightarrow A(2) \rightarrow B(2) \rightarrow A(3) \rightarrow B(3) = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$B(1) \rightarrow A(1) \rightarrow B(2) \rightarrow A(2) \rightarrow B(3) \rightarrow A(3) = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$B(1) \rightarrow A(1:2) \rightarrow B(3) \rightarrow A(3) = \begin{bmatrix} 0 \\ 0 \end{bmatrix} = A(1) \rightarrow B(1:2) \rightarrow A(2:3) \rightarrow B(3)$$

$$A(1:2) \rightarrow B(1) \rightarrow A(3) \rightarrow B(2:3) = \begin{bmatrix} 1 \\ -3 \end{bmatrix}, B(1:2) \rightarrow A(1) \rightarrow B(3) \rightarrow A(2:3) = \begin{bmatrix} -1 \\ 3 \end{bmatrix}$$

$$A(1:2) \rightarrow B(1:2) \rightarrow A(3) \rightarrow B(3) = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, B(1:2) \rightarrow A(1:2) \rightarrow \times = \begin{bmatrix} -1 \\ 0 \end{bmatrix}$$

14 حالت داشته‌یم

$$= \begin{bmatrix} 1 & 0 & 1 & 1 & -1 & 0 & 0 & 1 \\ -1 & 0 & -3 & -2 & 3 & -2 & 0 & -2 \end{bmatrix}$$

# توبه: طبق بند 4 ام دوک 2 در سرپوشی حذف شوند اگر نشوند حالت کلی زیاده باشد