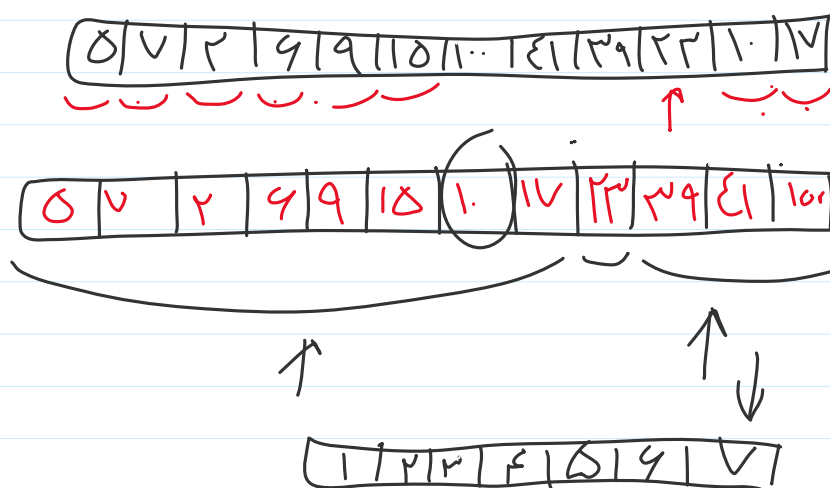


موضوع: مرتب سازی سریع تصادفی
حداکثر بر روی مرتب سازی تصادفی
* مرتب سازی خطی



مرتب سازی تصادفی: عنصر محور به صورت تصادفی انتخاب می کنیم

$$* متوسط زمان اجرا = O(n \log n)$$

* فرض: اعدادی که قرار است مرتب شوند n تا هستند.

$$X_{i,j} = \begin{cases} 1 & \text{اگر عدد } i \text{ در مرتب شده باشد} \\ 0 & \text{در غیر این صورت} \end{cases}$$

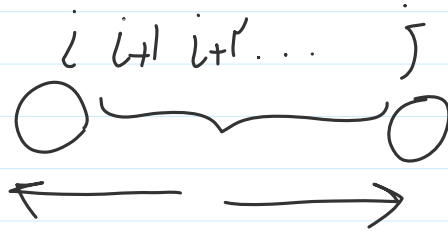
$$\text{تعداد کل مقادیر} = \sum_{i,j} X_{i,j}$$

$$P_{i,j} = \text{احتمال این که } i \text{ در } j \text{ مرتب شود}$$

$$P_{i,j} = ?$$

$$P_{ij} = ?$$

در چه صورتی i و j متابیه می شوند؟



* همیشه یک سمت متابیه کوردها هستند.

* i و j متابیه می شوند، اگر یکی از این در قبل از تمام اعداد $i \dots j$ به عنوان

کور انتخاب شود.

$$j > i$$

$$\frac{\mu}{j-i+1}$$

$$E(X_{ij}) = \frac{\mu}{j-i+1} \times 1 + 0 \times \left(1 - \frac{\mu}{j-i+1}\right)$$

$$= \frac{\mu}{j-i+1}$$

$$E\left(\sum_{i,j} X_{ij}\right) = \sum_{i,j} E(X_{ij})$$

$$= \sum_{i=1}^n \sum_{j=i+1}^n \frac{\mu}{j-i+1}$$

$$k < j-i$$

$$\begin{aligned}
 & \sum_{i=1}^n \sum_{k=1}^{n-i} \frac{1}{k+1} \\
 & \qquad \qquad \qquad \underbrace{\hspace{10em}}_{O(\log n)} \\
 & \underbrace{\hspace{15em}}_{O(n \log n)}
 \end{aligned}$$

حد پایین بر روی هر الگوریتم مرتب سازی مقایسه ای

$$O \leq O$$

~~XXXX~~

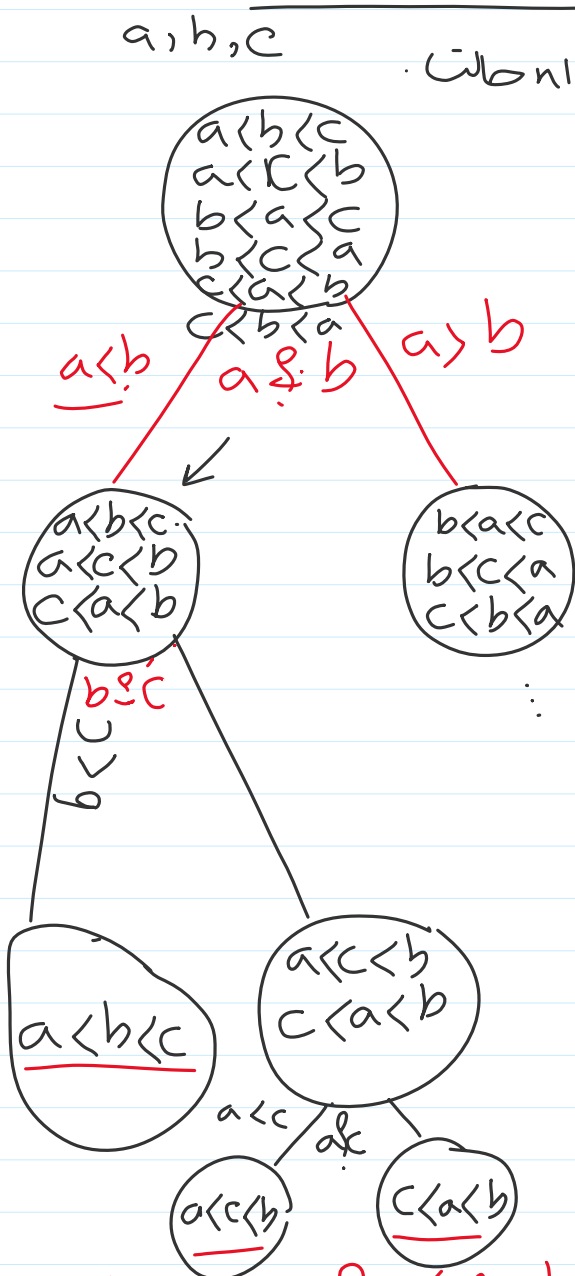
✖ هر الگوریتم مقایسه ای برای مرتب کردن n عنصر نیاز به $\Omega(n \log n)$ مقایسه دارد (در بدترین حالت) - (در حالت متوسط)

سوال ۳ عدد c و n داریم برای مرتب کردن این n عدد، حد اکثر به چند مقایسه نیاز داریم.

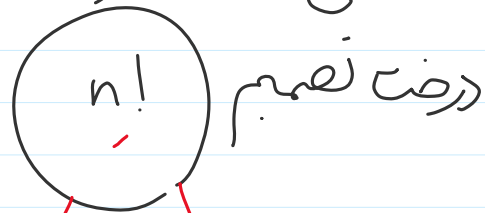
$a < b$
 $c a c b c$
 $c < b$

$c a b \rightarrow a ? c$
 $a c b$

← برای برش کردن $n!$ عنصر صفا نیاز به $\log n!$ مقایسه داریم



حالت کلی x_1, x_2, \dots, x_n ← $n!$ حالت



بزرگها، هر کدام یک جایست
 تعداد بزرگها $n!$

ارتفاع درخت $\log(n!) = \Omega(n \log n)$

← هر الگوریتم مرتب‌سازی مقایسه‌ای نیاز به $\Omega(n \log n)$ مقایسه دارد

← هو الگوریتم مرتب سازی مقایسه ای نیاز به $O(n^2)$ کار مقایسه دارد.

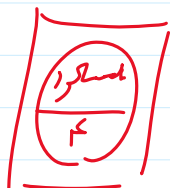
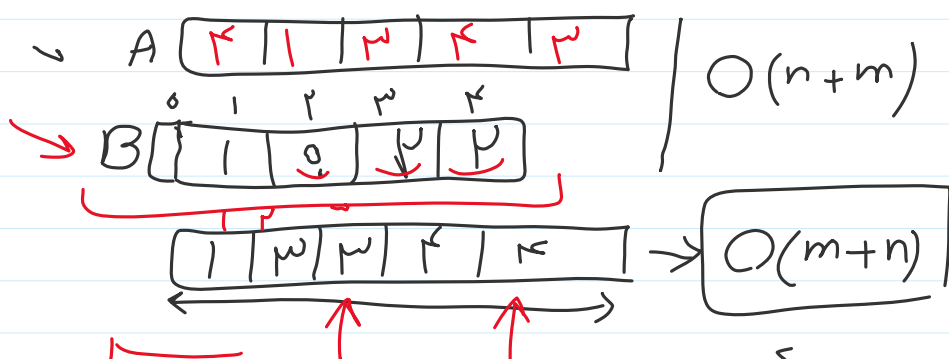
مرتب سازی خطی: (صفتی بر مقایسه نیست)

* مرتب سازی شمارشی Counting Sort

ورودی: n عدد می دانیم هم اعداد کمتر از m است.

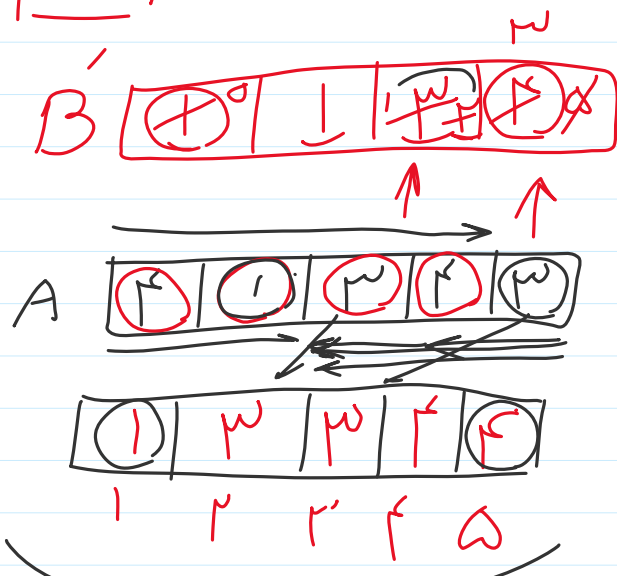
الگوریتم $O(m+n)$

$m=5$



* برای اعداد مشخصه قابل نشدیم

آخرین اندیسی $B[i]$
که کلید آفرای نگید



دلیل این که از انتها شروع
کردیم: پایداری

۱ ۱ ۱ ۳

اگر $m = O(n)$ باشد، کل زمان $O(n)$
