

(new) list Orderly()

for $i \leftarrow 0$ to $len-1$

print $arr[i]$

① findAt Index : trivial

② putAt Index : VERY restricted

③ insert (value) :

先"找到"

從頭 for $i \leftarrow 0$ to $end-1$

if $arr[i] < value$

插隊

從尾 for $i \leftarrow end-1$ to 0

if $arr[i] > value$

插隊

④ remove At Index (idx) :

"補位"

for $i \leftarrow idx+1$ to $end-1$

$arr[i-1] \leftarrow arr[i]$

$end \leftarrow end-1$

找到 + 插隊

can be combined here

(see textbook)

* about "找到" find Index Of (value)

find Record of (key)

score here

naive: Sequential Search

$len-1$ non-Consecutive

for $i \leftarrow 0$ to $end-1$ Consecutive

if $arr[i].score = key$

return $arr[i]$

return FAIL

Ordered 偷吃步

else if $arr[i].score < key$

return FAIL

* 偷吃步 ① for Ordered

if $\underset{\text{score}}{\text{arr}[i]} < \text{key}$ 後面都不用看了

偷吃步 ② for Ordered

if $\text{arr}[i] > \text{key}$ 前面都不用看了

$[\text{left}, \text{right}) \leftarrow [0, \text{end})$

while $\text{left} < \text{right}$ // not 空空如也

pick i

if $\text{arr}[i] = \text{key}$

return $\text{arr}[i]$ // 找到

else if $\text{arr}[i] < \text{key}$

find in ~~前面~~ // ①

$[\text{left}, i)$: still Consecutive & Ordered

|||
right $\leftarrow i$ left

else if $\text{arr}[i] > \text{key}$

find in ~~後面~~ // ②

$[i+1, \text{end})$: still Consecutive & Ordered

|||
left $\leftarrow i+1$ right

return FAIL

* "pick i ": ambiguous (violate definiteness)

• pick i within $[\text{left}, \text{right})$ better, but how?

• pick i ^{uniformly} randomly within $[\text{left}, \text{right})$?

• pick $i \leftarrow \text{left}$?

if lucky, 找到 or 偷一大步

else 沒偷到.

* Binary Search

pick $i \leftarrow \left\lfloor \frac{\text{left} + \text{right}}{2} \right\rfloor$ (usually named mid)

claim: after executing "pick i" $\lceil \log_2 \text{end} \rceil + 1$ times
right-left < 1 (空空如也)

lemma: every ^{non-found} updating after "pick i " decreases
right-left by at least half

case ① : right - left becomes

$$\left\lfloor \frac{\text{left} + \text{right}}{2} \right\rfloor - \text{left} \leq \frac{\text{left} + \text{right}}{2} - \text{left}$$
$$\quad \quad \quad \parallel$$
$$\quad \quad \quad \frac{\text{right} - \text{left}}{2}$$

case ② : right - left becomes

$$\text{right} - \left\lfloor \frac{\text{left} + \text{right}}{2} \right\rfloor - 1$$
$$\leq \text{right} - \left(\frac{\text{left} + \text{right}}{2} - 1 \right) - 1$$
$$= \frac{\text{right} - \text{left}}{2}$$

so after $\lceil \log_2 \text{end} \rceil + 1$ times of "pick $\bar{u}$ "

$$\begin{aligned} \text{right} - \text{left} &\leq \frac{\text{end} - 0}{2^{\lceil \log_2 \text{end} \rceil + 1}} \\ &\leq \frac{\text{end}}{2^{\log_2 \text{end} + 1}} \\ &\leq \frac{1}{2} \\ &< 1 \end{aligned}$$

$\Rightarrow$ Binary Search terminates after at most $\lceil \log_2 \text{end} \rceil + 1$ iterations