Introduction to Algorithms

Mong-Jen Kao (高孟駿)

Tuesday 10:10 – 12:00

Thursday 15:30 – 16:20

Binary Search

Find the ***boundary of 0-1*** in a ***0-1 monotone sequence fast***.

Two Typical Scenarios

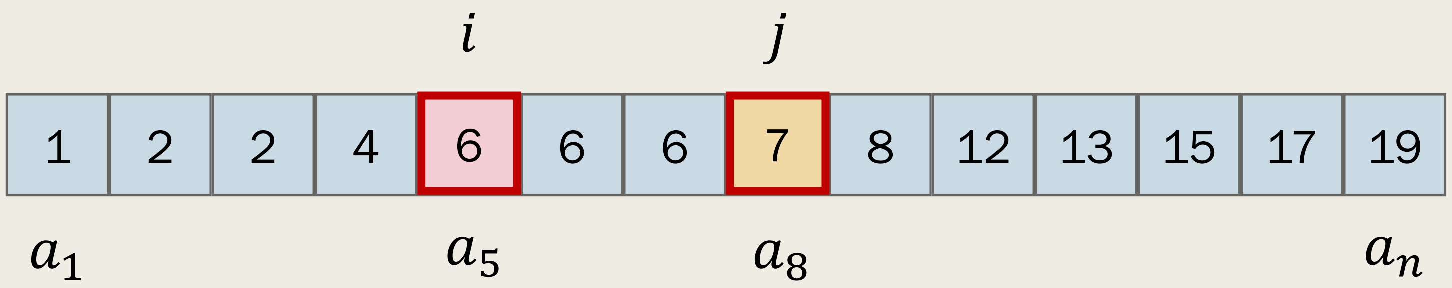
- Given a sequence of numbers $a_1, a_2, \ldots, a_n$ that are ***sorted*** in *non-descending order*.

For *a given value* k,

- Find the smallest index i such that $a_i \nless k$.
- Find the smallest index j such that $a_j > k$.

the first element $\geq k$.

the first element $> k$.



For $k = 6$,

$i = 5$

$j = 8$.

Two Typical Scenarios

- Given a sequence of numbers $a_1, a_2, \ldots, a_n$ that a
non-descending order.

For *a given value* k,

- Find the smallest index i such that $a_i \not< k$.
- Find the smallest index j such that $a_j > k$.

$j - i$ is the number of times k appears.

the first element $\geq k$.

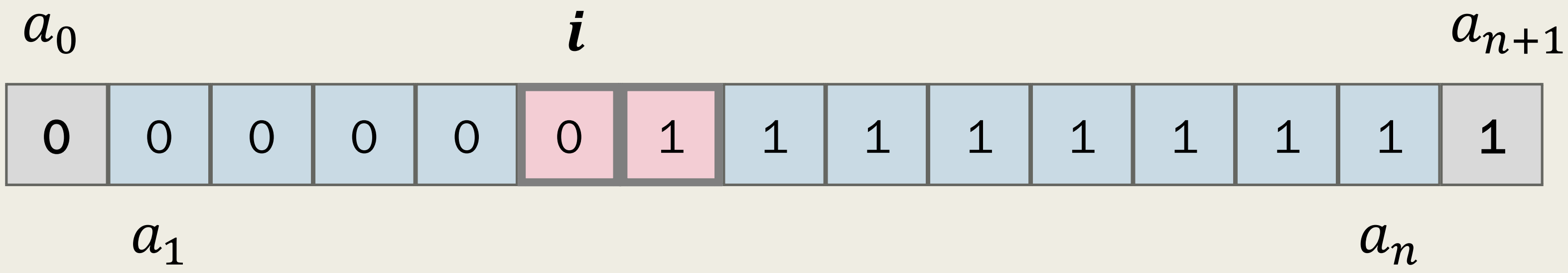
the first element $> k$.

				i			j						
1	2	2	4	6	6	6	7	8	12	13	15	17	19
a_1				a_5			a_8						a_n

For $k = 6$,
$i = 5$
$j = 8$.

The General Scenario

- Given a 0-1 sequence $a_1, a_2, \dots, a_n$ *sorted in order*, *further assume* that $a_0 = 0$ and $a_{n+1} = 1$.
  - Find the index i such that $a_i \neq a_{i+1}$, i.e., identify the boundary of 0 and 1.



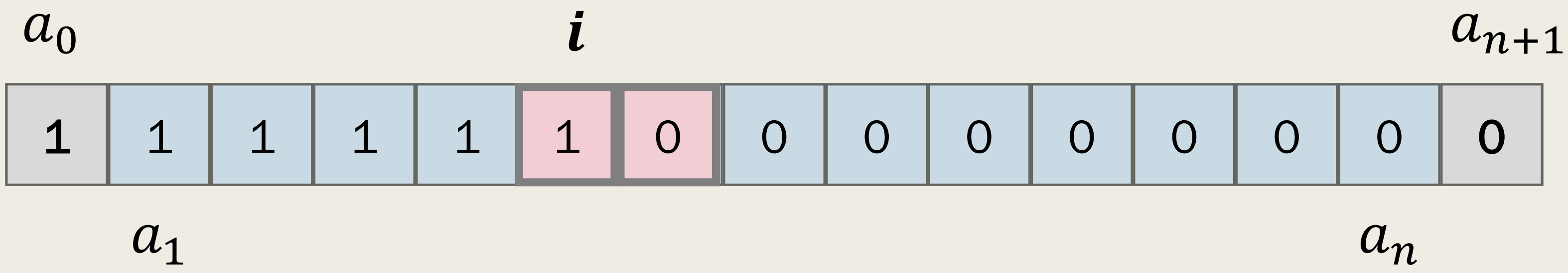
Alternative (Equivalent) Scenario

- Given a 0-1 sequence $a_1, a_2, \dots, a_n$ *sorted in order*,

  further assume that $a_0 = 0$ and $a_{n+1} = 1$.

  – Find the index i such that $a_i \neq a_{i+1}$,

    i.e., identify the boundary of 1 and 0.



Conversion to the General Scenario

- The first search problem can be converted to the general form.
  - Find the smallest index i such that $a_i \nless k$.

For $k = 6$,

$-\infty$	1	2	2	4	6	6	6	7	8	12	13	15	∞
	a_1				a_5							a_n	

For each element a_i, we ask

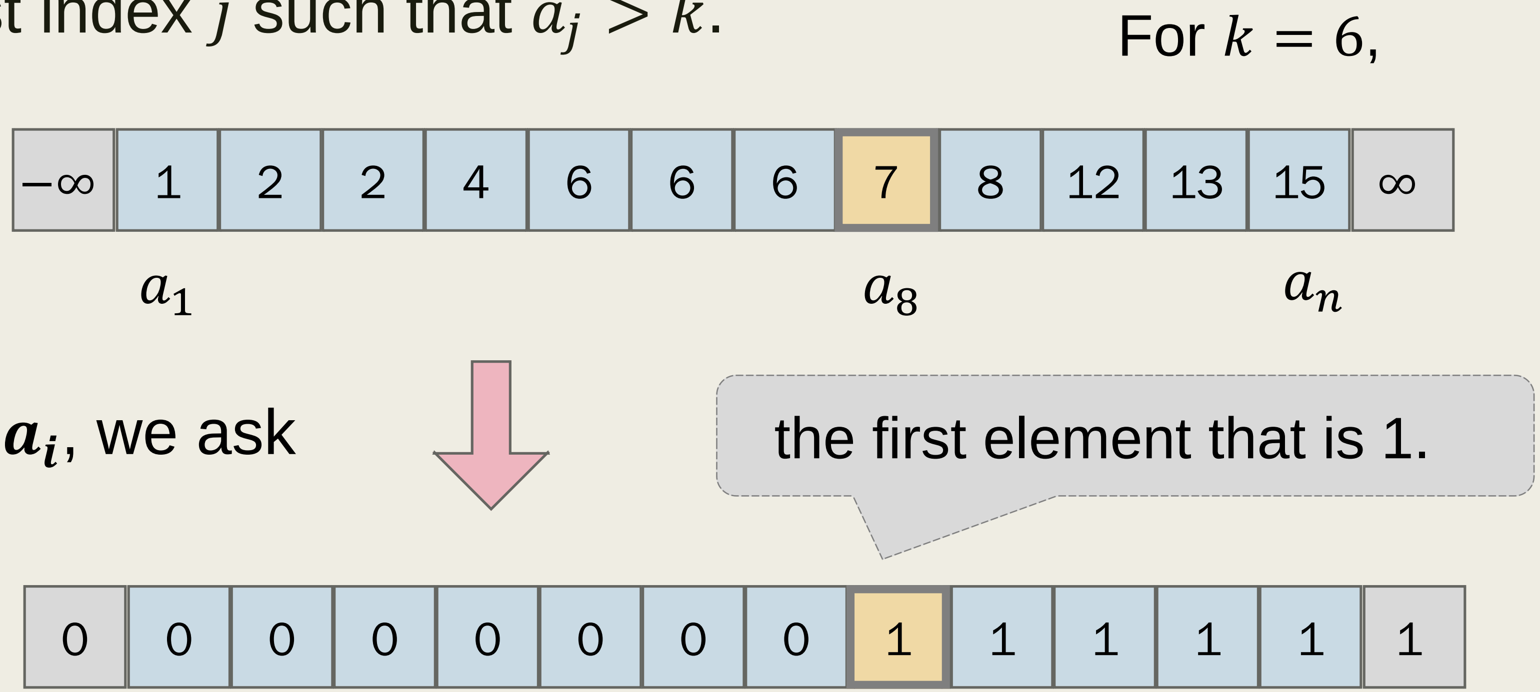
- **Is $a_i < k$?**

the first element that is 0.

1	1	1	1	1	0	0	0	0	0	0	0	0	0

Conversion to the General Scenario

- The second search problem can be converted to the general form, too.
  - Find the smallest index j such that $a_j > k$.



For each element a_i, we ask

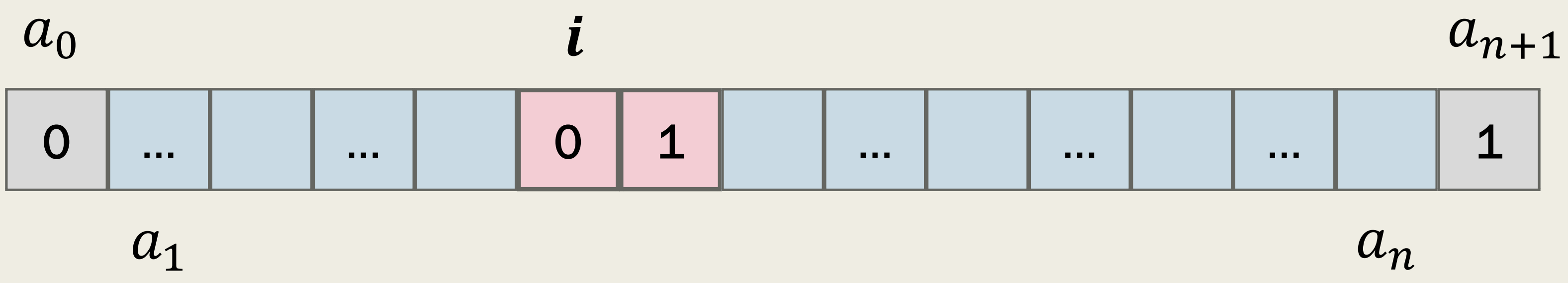
- **Is $a_i > k$?**

Binary Search on 0-1 Sequence

Find the ***boundary of 0-1*** in a ***0-1 monotone sequence fast***.

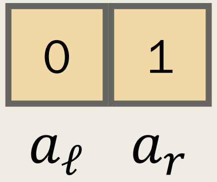
Problem Scenario

- Let $a_1, a_2, \dots, a_n$ be a 0-1 sequence of interests.
  - We further assume that $a_0 = 0$ and $a_{n+1} = 1$.
  - Find the index $i \in \{0,1,\dots,n\}$ such that $a_i = 0$ and $a_{i+1} = 1$.



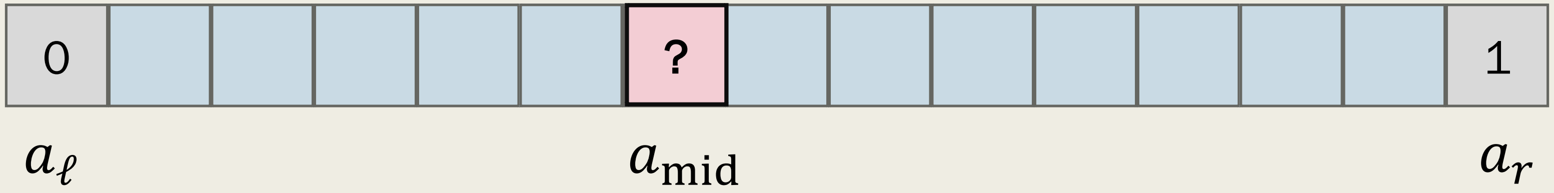
Let $a_1, a_2, \ldots, a_n$ be a 0-1 sequence. Assume that $a_0 = 0$ and $a_{n+1} = 1$.

- Given two indexes $\ell < r$ with $a_\ell = 0$ and $a_r = 1$, find the index $i \in [\ell, r-1]$ such that $a_i = 0$ and $a_{i+1} = 1$.
  - If $r - \ell$ is 1, then we're done.



  - Otherwise, $r - \ell > 1$.

    Take $\mathrm{mid} \coloneqq \lfloor (\ell + r)/2 \rfloor$ and inspect a_{mid}.



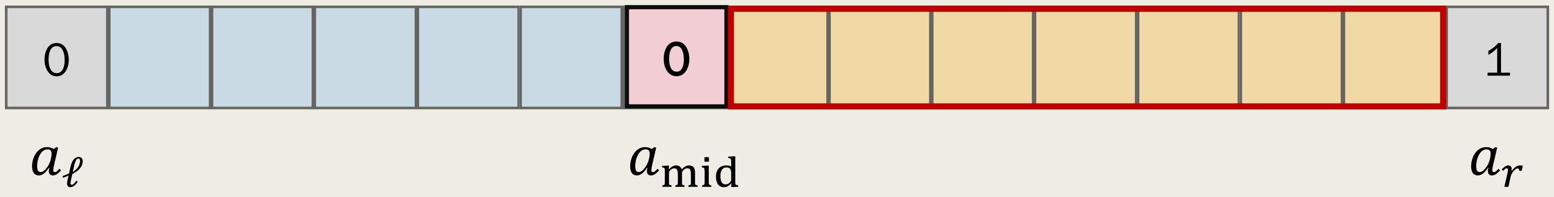
Let $a_1, a_2, \ldots, a_n$ be a 0-1 sequence. Assume that $a_0 = 0$ and $a_{n+1} = 1$.

- Given two indexes $\ell < r$ with $a_\ell = 0$ and $a_r = 1$,
  find the index $i \in [\ell, r-1]$ such that $a_i = 0$ and $a_{i+1} = 1$.

  – Take $\mathrm{mid} \coloneqq \lfloor (\ell + r)/2 \rfloor$ and inspect a_{mid}.

    - If $a_{\mathrm{mid}} = 0$, then the answer is in the right-hand-side.

      We have a ***recursive problem*** on (mid, r).



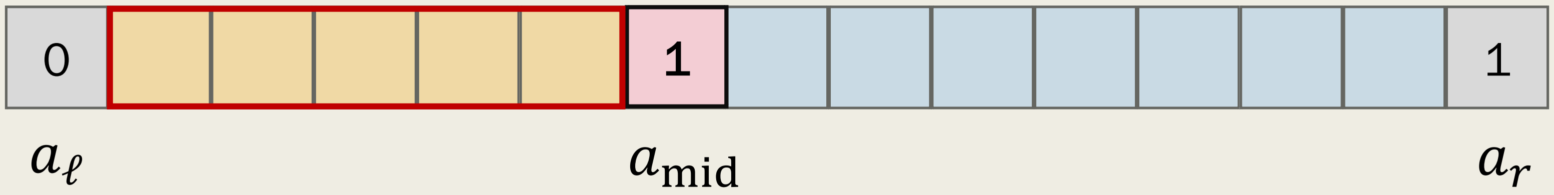
Let $a_1, a_2, \ldots, a_n$ be a 0-1 sequence. Assume that $a_0 = 0$ and $a_{n+1} = 1$.

- Given two indexes $\ell < r$ with $a_\ell = 0$ and $a_r = 1$,
  find the index $i \in [\ell, r-1]$ such that $a_i = 0$ and $a_{i+1} = 1$.

  – Take $\mathrm{mid} \coloneqq \lfloor (\ell + r)/2 \rfloor$ and inspect a_{mid}.

  - If $a_{\mathrm{mid}} = 1$, then the answer is in the left-hand-side.

    We have a ***recursive problem*** on (ℓ, mid).



- BinarySearch(L, R) - To search the 0-1 sequence $a_L, \ldots, a_R$

A. $\ell \leftarrow L - 1$.
$r \leftarrow R + 1$.

B. While $r - \ell > 1$, do the following.

a) $\mathrm{mid} \leftarrow \lfloor (\ell + r)/2 \rfloor$.

b) If a_{mid} is 0, set $\ell \leftarrow \mathrm{mid}$.
Otherwise, set $r \leftarrow \mathrm{mid}$.

C. Report (ℓ, r).

Correctness of Binary Search

- In step (Ba),
  we always have $L \leq \mathrm{mid} \leq R$.
  - When $\ell < r - 1$, we have
  $$\ell < \lfloor (\ell + r)/2 \rfloor < r\,.$$
- The answer to be searched is always contained within $[\ell, r]$
  in the beginning of the while loop in step (B).

Time Complexity

- The running time of this algorithm can be described by the following recurrence.

$$T(n) = \begin{cases} \Theta(1), & n \leq 2, \\ T(\lfloor n/2 \rfloor) + \Theta(1), & n > 3. \end{cases}$$

  – Solving the recurrence, we obtain $T(n) = \Theta(\log n)$.

- LowerBound($A[1 \ldots n], k$) - A[1…n] sorted in non-descending order.
  Find the smallest i such that $A[i] \geq k$

A. $\ell \leftarrow 0.$
   $r \leftarrow n + 1.$

B. While $r - \ell > 1$, do the following.
   a) $\text{mid} \leftarrow \lfloor (\ell + r)/2 \rfloor.$
   b) If $a_{mid} < k$, set $\ell \leftarrow \text{mid}.$
      Otherwise, set $r \leftarrow \text{mid}.$

C. If r equals $n + 1$, then report “No such element”.
   Otherwise, report r.

- UpperBound($A[1 \dots n], k$) - A[1…n] sorted in non-descending order.
  Find the smallest i such that $A[i] > k$

A. $\ell \leftarrow 0$.
   $r \leftarrow n + 1$.

B. While $r - \ell > 1$, do the following.
   a) $\text{mid} \leftarrow \lfloor (\ell + r)/2 \rfloor$.
   b) If $k < a_{mid}$, set $r \leftarrow \text{mid}$.
      Otherwise, set $\ell \leftarrow \text{mid}$.

C. If r equals $n + 1$, then report “No such element”.
   Otherwise, report r.