Lower Bounds for Searching

Model:

- A known set S of elements is stored. In particular, their order is known.
- An unknown element x is presented at query time.
- We assume that all elements of S are distinct and that $x \in S$ (this is enough for lower bounds).
- The objective is to find the (unique) $s \in S$ for which $s = x$.
- The basic operation is a comparison of $s \in S$ with x.
- The cost of an algorithm is the number of such comparisons.

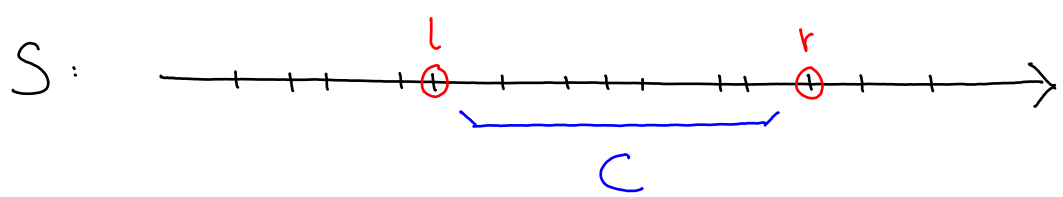
Adversary argument:

At any time let $\tilde{S} \subseteq S$ be the elements which have been compared to x, and let

$$r = \min \{ s \in \tilde{S} \mid x < s \}$$

$$l = \max \{ s \in \tilde{S} \mid x > s \}$$

$$C = \{ s \in S \mid l < s < r \}$$



C is the set of current candidates for x.

Search algorithm cannot stop before $|C| = 1$.

Adversary giving answers to comparisons has been consistent so far if $|C| \geq 1$.

Define adversary as follows: on comparison of x with $s \in S$ it answers

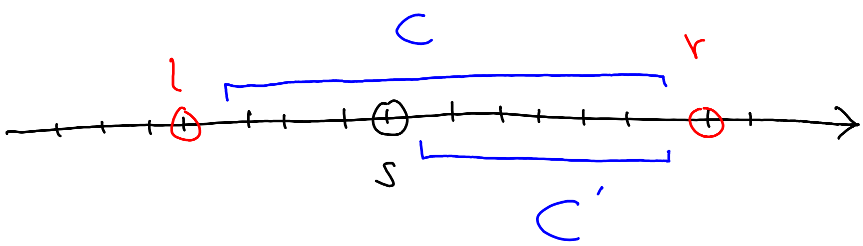
$$x < s \quad \text{if } s \geq r.$$
$$x > s \quad \text{if } s \leq l.$$

This does not change r and l. Else it answers

$$x < s \quad \text{if } s \text{ is in right half of } C \quad (r \text{ moves}).$$
$$x > s \quad \text{if } s \text{ is in left half of } C \quad (l \text{ moves}).$$

Let C' be the set of candidates after the comparison.

Then $|C'| \geq \frac{|C|-1}{2}$, as illustrated below for the last case:



Thus, after k comparisons we have

# comp.	$\lvert C \rvert$
0	$= N$
1	$\geq \frac{1}{2}(N-1) = \frac{1}{2}N - \frac{1}{2}$
2	$\geq \frac{1}{2}\left(\left(\frac{1}{2}N - \frac{1}{2}\right) - 1\right) = \frac{1}{2^2}N - \left(\frac{1}{2} + \frac{1}{2^2}\right)$
⋮	
k	$\geq \frac{1}{2^k} \cdot N - \left(\frac{1}{2} + \frac{1}{2^2} + \frac{1}{2^3} + \cdots + \frac{1}{2^k}\right)$

As $\frac{1}{2} + \frac{1}{2^2} + \cdots + \frac{1}{2^k} < \frac{1}{2} \cdot \sum_{i=0}^{\infty} (\frac{1}{2})^i = \frac{1}{2} \cdot \frac{1}{1 - 1/2} = \frac{1}{2} \cdot 2 = 1$,

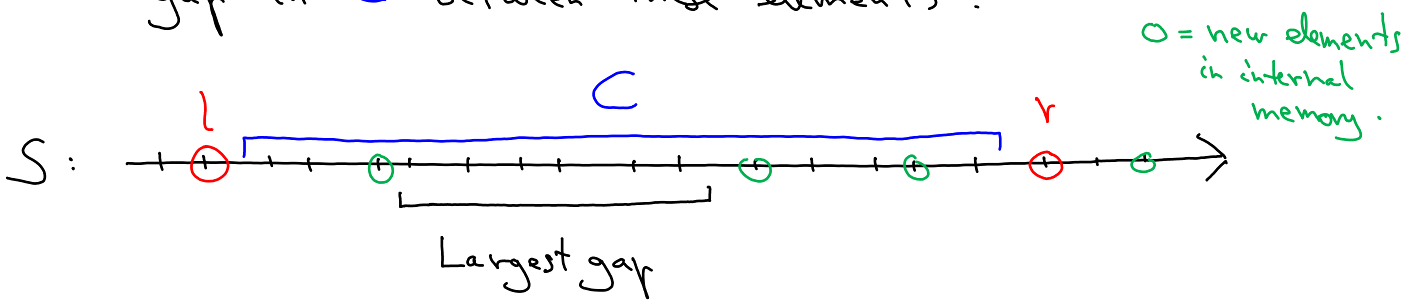
we have $|C| > \frac{1}{2^k} \cdot N - 1$ after k comparisons. As the algorithm cannot give a correct answer unless $|C| = 1$ (else adversary can claim another answer correct than the one given by the algorithm), we see

$$
\begin{aligned}
& 1 > \frac{1}{2^k} \cdot N - 1 \\
\Updownarrow \quad & 2 > \frac{1}{2^k} \cdot N \\
\Updownarrow \quad & 2^{k+1} > N \\
\Updownarrow \quad & k+1 > \log_2 N \\
\Updownarrow \quad & k > \log_2(N) - 1
\end{aligned}
$$

As k is an integer, we see $k = \lfloor \log_2 N \rfloor$ comparisons are necessary for comparison based searching.

We now consider comparison based searching in the I/O-model. Here, comparisons can only take place between elements in internal memory, and the cost is the number of I/O's performed.

Each I/O brings at most B new elements of S into internal memory. The adversary will choose an answer to all comparisons possible between x and these elements by declaring x lying in the largest gap in C between these elements:



Hence, $|C'| \geq \frac{|C| - B}{B+1}$, as there are at most $B+1$ gaps in C. So we have

# I/o's	$\lvert C \rvert$
0	$= N$
1	$\geq \frac{N-B}{B+1} = \frac{N}{B+1} - \frac{B}{B+1}$
2	$\geq \frac{N/_{B+1} - B/_{B+1} - B}{B+1} = \frac{N}{(B+1)^2} - B\left(\frac{1}{(B+1)^2} + \frac{1}{B+1}\right)$
$\vdots$	
k	$\geq \frac{N}{(B+1)^2} - B\left(\frac{1}{(B+1)^k} + \dots + \frac{1}{(B+1)^2} + \frac{1}{B+1}\right)$

As $\frac{1}{(B+1)^k} + \dots + \frac{1}{(B+1)^2} + \frac{1}{B+1} < \frac{1}{B+1} \cdot \sum_{i=0}^{\infty} \left(\frac{1}{B+1}\right)^i$

$$= \frac{1}{B+1} \cdot \frac{1}{1 - \frac{1}{B+1}} = \frac{1}{B+1} \cdot \frac{1}{\frac{B}{B+1}} = \frac{1}{B+1} \cdot \frac{B+1}{B} = \frac{1}{B}$$

we have $|C| > \frac{N}{(B+1)^k} - 1$ after k I/o's.

Hence

$$1 > \frac{N}{(B+1)^k} - 1$$

$$\Updownarrow$$

$$2 > \frac{N}{(B+1)^k}$$

$$\Downarrow$$

$$(B+1)^{k+1} > N$$

$$\Updownarrow$$

$$k > \log_{B+1}(N) - 1$$

So at least $\lfloor \log_{B+1}(N) \rfloor$ I/O's are needed.