LECTURE 6

BINARY SEARCH

Po Hao Chen

June 4, 2023

BOSTON UNIVERSITY

Generic Binary Search

Given a sorted array s.t

$a_i \leq a_{i+1}$

Example: 2,3,4,8,10,12

Find x = 10

```
L = 0, R =  n- 1
while(L <= R)
        M = L + (R-L) / 2
        if( a[M] == X )
                "Found"
        else if( a[M] < X)
                L = M + 1
        else
                R = M – 1
"Not Found"
```

Find Closest Element

Find the first element greater or equal to x

Example: 1,2,5,18,19,20

x = 3

```
L = 0, R  = n -1
ans = -1
while(L <= R)
      M = L + ( R – L ) / 2
      if a[M] >= target
            ans = a[M]
            R = M – 1
      else
            L = M + 1
return ans
```

Universal Binary Search

$16 = ? * ?$ You can take the square root of the LHS.

Or you can "guess" and verify.

Formulate our problem such that our "answer" lies within some [L,R].

Universal Binary Search

$$f(x) = \begin{cases} 1 & \textit{satisfies } f \\ 0 & \textit{otherwise} \end{cases}$$

Maintain f(l) = 0, f(r) = 1 (Depending on the problem)

Implementation

Step 1: Choose your L,R

Step 2: Implement f(x)

Step 3: Apply the framework

```
L = 0
R = "upper bound value"

while (L <= R)
    m = L + ( R – L) / 2
    if( f (m) )
        R = m – 1
    else
        L = m + 1

Return R + 1
```

Rotated Array

Find the pivot point of the following array after rotation.

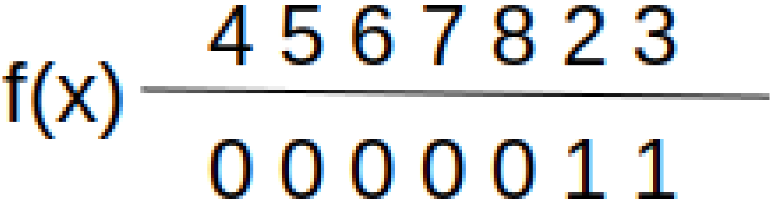
Before: 2,3,4,5,6,8

After: 4,5,6,8,2,3

Solution

L = 0, R = n-1

f(x): checks if x is less than the first element.



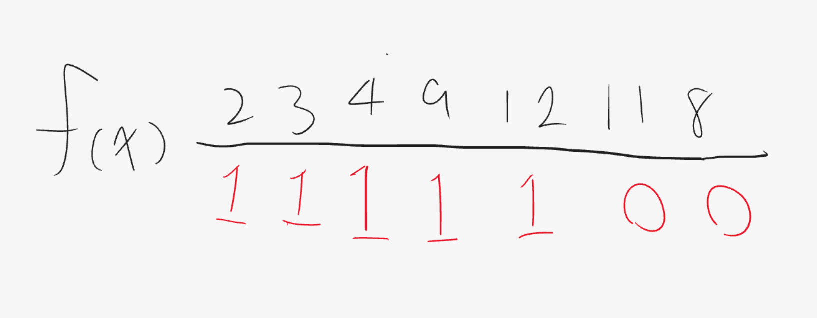
Finding Peak

Find the maximum element in the sequence
2,3,4,5,6,9,12,11,8,6,4,1

Solution

f(l) = 1, f(r) = 0

f(x): check if $x > x - 1$ or is the first element

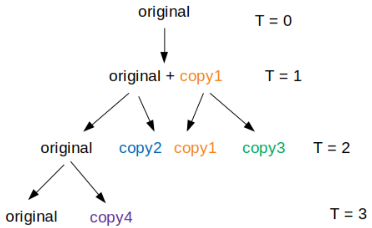


We will modify our template.

Copy Machine

Given an original copy, make n copies. You have two copier, the first one finishes the job in x seconds, the other in y seconds. Find the minimum required time.

Example:
4 1 1
It takes 3 seconds
5 1 2
It takes 4 seconds



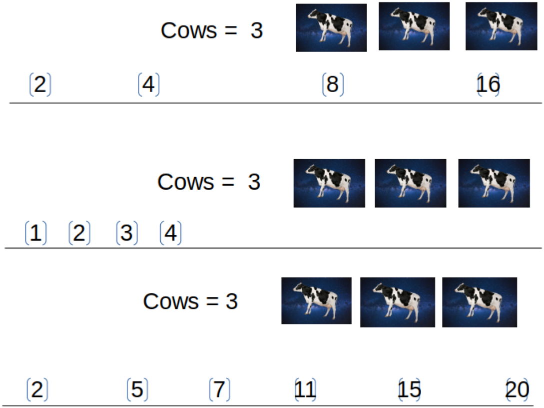
Copy Machine

- Search for the minimum time sufficient for n copies to be made.
- f(t) = Can you make n copies in t time using the provided machines?

```
f(T)
  T -= min(x,y)
  if(T < 0) return false;
  Else return (time / x + time / y + 1 ) >= n
```

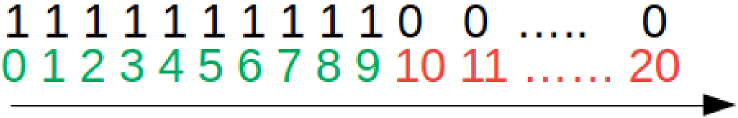
max(min()) Problem

There are n stalls in a straight line. Place k cows into the stalls such that their minimum distance is maximized.



Approach

- We are looking for the minimum "distance", thus, we can search a range of candidate number and verify its validity.
- To maximize it, we will find the last valid distance.



- What verifies the validity of a candidate x?
  Check each segment, if it's greater or equal to x. Add to count.
  When count equals $|cows|$, it's valid.

Solution

9 is the maximum possible distance between the two closest cows.