Binary Search on real numbers

Given a rope of different lengths $a[i] \rightarrow$ length of the rope

$\rightarrow 2.1$
$\rightarrow 3.4$
$\rightarrow 5.9$
$\rightarrow 2.1$
$\rightarrow 3.6$

$[2.1, 3.4, 5.9, 2.1, 3.6]$

You need to cut k pieces of same length

Task is to find the maximum possible length of pieces.

for a rope $\textcircled{5}$, It is good or not if, if we can k pieces of length x

max value of $\textcircled{5}$ (budget $\rightarrow$ $\textcircled{5}$)

(good) $\rightarrow$ Time/2, if I can cut k pieces of length $\textcircled{5}$.

Now $\textcircled{5}$ is good, then for any length less than $\textcircled{5}$ we can also cut k pieces $\rightarrow -1$

$f(x) \geq k \Rightarrow f(x) = 1$

$x - 1$

$x - 1$ $\rightarrow$ Explicit value

Solve using Binary search:

① Search space

l, r

$\rightarrow$ read rope.

while $(l < r)$ $\{$

while $(\boxed{\quad} - l) > 10^{-6}$ $\}$

$m = (l + r)/2$;

if $(f(m) == 1)$ $\{$

$l = m$;

$\}$

else

$r = m$;

$\}$

for real rng numbers can be infinitely close to each other and still not overlap.

$\rightarrow$ This value can be hard too. we necessarily can read.

while $(r - l) > 10^{-6}$

10^{-1}

Too big $\varepsilon \rightarrow$ not accurate

Too small $\varepsilon \rightarrow$ lots of 10^{-100}

$l = 3.81356 \ldots\ldots 57 \cdot 10^{-30}$

$r = 3.81352 \ldots\ldots 61 \cdot 10^{-30}$

very deep to find real value.

So, better way to do this is

$\boxed{\text{fixed no of iterations}}$

if $1/r \approx 10^{-3}$

accy $\approx 10^{-3}$

we need atleast $\log\left(\frac{\text{range/acc}}{\text{eps}}\right)$

Iterations

$\log(10^{3}) \approx 100$

So, just to be safe, 100 iterations.

for $(i = 0 \rightarrow 100)$ $\{$

$\}$

gen1 function

$\rightarrow$ We need to build a $f(x)$ that tells if it is possible to cut k pieces of length $\textcircled{5}$.

$\rightarrow$ Easy way

$\textcircled{5}$ Iterate well all the ropes check the way you if pieces that can be cut.

$\rightarrow 8.2 \rightarrow 4$ pieces $\quad 20 + 5 + 1 = 8.2$
$\rightarrow 4.3 \rightarrow 3$ pieces
$\rightarrow 9.2 \rightarrow 3$ pieces
$\rightarrow 5.7 \rightarrow 2$ pieces

output $\rightarrow 5.7$

Linear function $\rightarrow$ Binary search ✓
Array sorted $\rightarrow$ Binary search ✗
$\rightarrow$ $\frac{\text{sort}}{\text{desc}}$

$\rightarrow 1\ 2\ 3\ 3\ 3\ 4\ 4\ \textcircled{5}\ 5\ 5\ 6\ 5\ 5\ 5\ 4\ 2\ 2\ 2$

Monotonic sequence

$\hookrightarrow$ numbers appear in decreasing order, then suddenly start to increase.

$\textcircled{5} \rightarrow$ max number in $O(\log n)$

(output) $\rightarrow$ $\textcircled{5}$ $\quad$ 3 $\quad$ $\textcircled{5}$ $\quad$ 3 $\quad$ $\textcircled{5}$ (not all no)

L $\quad$ mid $\quad$ R

$\max[mid] == 5$

$\rightarrow$ problem form to be solved.

Binary search not dead m.

Binary search applications

$\rightarrow$ embedded function to determine the max or min value of a function

$\rightarrow$ monotonic function have single output value.

$[a, b]$

determine $\textcircled{5}$ $\quad$ $\max(a) \rightarrow$ $\textcircled{5}$ $\rightarrow$ $\frac{\textcircled{5}}{2}$ $\rightarrow$ $\frac{\textcircled{5}}{4}$

$m = \frac{a+b}{2}$

$a + m + m + b$

$[l_1, r_1]$

$\rightarrow$ check differentiation

$\rightarrow$ modified binary search

embedded function

$f(x) = -x^2 + 6x + 3$

where

$a = 1$,
$b = 6$,
$c = 3$

determine the value of x where $f(x)$ will be max.

search ( ) $\{$

double l = inf;

double r = sup;

for $(int\ i = 0; i < iter; i++)$ $\{$

double mid1 = $(l + (r - l)/3)$;

double mid2 = $(r - (r - l)/3)$;

if $(f(m1) > f(m2))$ $\{$

$r = m2$

else

$l = m1$

$\}$

double $x = \frac{l+r}{2}$;

return $f(x)$;

$\}$

double $f($double x$)$ $\{$ $\quad$ $-x^2 + 2x + 3$;

return $-x^2 + 2x + 3$;

$\}$

search in rotated sorted array

$\underline{Q}$ Have a an integer array (sorted) ascending order

array rotated at an unknown point

$\max[0] \ldots\ldots\ldots\ldots\ldots\ldots \max[n-1]$

$\max[k], \max[k+1] \ldots \max[n-1], \max[0], \ldots \max[k-1]$

$\rightarrow [0, 1, 2, 4, 5, 6, 7]$

rotate $\rightarrow$ $\textcircled{4}$ (pivot)

$\rightarrow [4, 5, 6, 7, 0, 1, 2]$

Given the array, reflect the rotation and return the index of target, if it is in array

$\rightarrow$ $\frac{\text{target}}{\text{length}}$

arr $\rightarrow [4, 5, 6, 7, 0, 1, 2]$

target $\rightarrow 0$

output $\rightarrow 4$

$\underline{Eg}$: arr $\rightarrow [4, 5, 6, 7, 0, 1, 2]$

target $\rightarrow 3$

output $\rightarrow -1$

One-pass Binary search

4 5 6 7 0 1 2

① start $\rightarrow 0$
end $\rightarrow n-1$

② find mid (start + end)

③ Decide based on mid

$\rightarrow$ $\max[mid] > target$

$\max[mid] \rightarrow$ mid

$\rightarrow$ mid value is smaller than point, so it's array

$\rightarrow$ rotation index $\textcircled{0}$