Searching
Tuesday, 30 August 2022 8:25 PM

Q: Guess the word:

10101110111000001 + 0000

Binary — Searching

y = 25 — Linear

Search

Agenda:

1. Python Pattern Question
2. DSA → when? why? (motivation)
3. Searching
4. Linear Search
5. Binary Search.

Q: n = 5

Print:

```
1 1 1 1 1
1 5 5 5 1
1 5 1 5 1
1 5 5 5 1
1 1 1 1 1
```

→ i, j ∈ (1, n)

→ loops, nested loops

→ 2D DSA

→ Grounds & dashboards.

4 (2, 2)

(n, n)

Observation

↳ we know small so: more towards the case.

from left
→ i, j

from top
→ i, j

from bottom
→ i, n-i

from right
→ n-j, n-i

n = 5-i

5-i-1

= 4

max(i, j, n-j, n-i) + 1

i, j = (2, 2)

⇒ max(i, j, n-i-j, n-i-j) + 1

max(2, 2, 3, 2) + 1

3 + 1

4

1	1	1	1	1
1	2	2	2	1
1	2	3	2	1
1	2	2	2	1
1	1	1	1	1

Hint (optional)

n = 4

(Rings)

Data structures & Algorithms (DSA)

Q: What is a data structure?

A → store data, strings, lists, organize data

Q2: Algorithm?

→ set of instructions to a specific order.

"Algo" → make a cup of tea.

→ ask, heat water, tea, milk

→ add sugar / no sugar / ginger.

→ add water

→ let brew for 3 min.

→ Tea is ready.

Q: Why DSA in long?

→ make our code efficient

→ stuck to optimal solution.

(P) (S)

Searching.

→ Linear Search

→ Binary Search.

Why Searching?

Google

↳ "Search"

Amazon Product Search

Linear Search.

Return-Idx = [1, 8, 2, 1, 3, 3, 3] [0-n]

target = 3 → found at index 5

target = 6 → not present → any in case ele.

n = len(arr)

for i in range(n):

if arr[i] == target

return "found at index" + i

return "not found"

(Optional)

Binary Search

Q → ↳ is works only on "sorted array."

an array where elements are ordered in "asc / desc".

[1, 5, 3, 9, 11] ✓

[3, 5, 8, 4, 4, 2, 1] ✗

[5, 3, 1, -2] ✓

arr = [2, 3, 5, 7, 9, 11, 13]

target = 11

start, end

mid = (start + end)//2

(0 + 7)//2

= 3

if arr[mid] == target:

return "found" + mid

dif target > arr[mid]

start = mid + 1

mid = (s+e)//2

if arr[mid] == target:

return "found"

elif target > arr[mid]

start = mid + 1

else

end = mid - 1

arr = [2, 3, 5, 7, 9, 11, 13]

target = 5

start, end

mid = (s+e)//2

= (0+6)//2

= 3

if arr[mid] == target:

return "found"

elif target > arr[mid]

start = mid + 1

else

end = mid - 1

arr = [2, 3, 5, 7, 9, 11, 13]

target = 6

s, mid, e

(s+e)//2

= 3

(s = mid+1)

mid = (s+e)//2

= (0+6)//2

= 3

(s = mid+1)

while (s ≤ e)

size

[3, 5, 3, 9]

s, mid, e

mid = (s+e)//2

(0+3)//2 → 1.5 → 1

[1, 4, 9, 10, 15, 19]

s, e

target = 6

DSA

→ Binary Search

→ Python Tips/Tricks / Curiosity

"Ex" ✗

DSA

"Python is slow"

→ Fast (C)

→ "Python is fast"

a → [1, 2, 3, 4]

b = a

b[0] = 100

c = a.copy()

c → [1, 2, 3, 4]

Plan

[3.8, 4.2, 5.6, 9.1]

→ Binary Search

["a", "b", "c", "d", "e", "f"]

"c" < "g"

mid = (s+e)//2

["Mango", "Apple", "Banana", "Kiwi"]

"Apple" > "Mango"

[' ', 'B', 'Fruit', 'Apple',]

Your

Mom

n = 7

0, 1, 2, 3, 4, 5, 6

s, mid, e

mid = (s+e)//2 = 3

n = 8

0, 1, 2, 3, 4, 5, 6, 7

s, mid, e

mid = (s+e)//2

= 3.5 → 3

[5, 7] (mid) → 148

s = mid + 1

[7, 8]

200

s = mid

s = mid

[ ]

[7, 8, 3, 2, 0, -5]

arr[3] < arr[-5] < arr[1] < arr[0]

arr[::-1]

[ ]

all the even numbers