Computer Science Fundamentals

Seminar title: Algorithms | Binary search

Lecturer: Olga Yugay

Agenda

- Lecture review
- Algorithms
- Binary search

Q&A

What is algorithm?

How do you represent algorithm?

Task 1

Note the three phases of the computer problem-solving model.

1. Algorithm development phase
2. Implementation phase
3. Maintenance phase

Task 1 continued

Match the phases with definitions.

Algorithm Development	The phase includes coding (translating the algorithm into a computer language) and testing (compiling and running the program).
Implementation	The phase involves using the program and modifying the program to add functionality or correct errors.
Maintenance	The phase includes analysis (understanding the problem), proposed solution (logical sequence of solution steps), and testing (following algorithm).

Task 1 answer

Match the phases with definitions.

Algorithm Development	The phase includes analysis (understanding the problem), proposed solution (logical sequence of solution steps), and testing (following algorithm).
Implementation	The phase includes coding (translating the algorithm into a computer language) and testing (compiling and running the program).
Maintenance	The phase involves using the program and modifying the program to add functionality or correct errors.

Task 2

Complete the following quiz:

https://quizizz.com/join/quiz/6190bcfbaa5222001ef6404d/start?studentShare=true

Review binary search:

Given list of numbers find number 5

1, 2, 5, 7, 8, 10, 50, 60, 71, 77, 78, 79, 80

x=5

Review binary search:

Given list of numbers find number 5

1, 2, 5, 7, 8, 10, 50, 60, 71, 77, 78, 79, 80

x=5

1) Midpoint is 50

To identify midpoint:

1. Count the total number of number entries (which is 13 in this example)
2. Divide the total number by 2 (in this example, 13/2=6.5
3. Round the result 6.5 ≈ 7, 7th number is the midpoint

Review binary search:

Given list of numbers find number 5

1, 2, 5, 7, 8, 10, ~~50, 60, 71, 77, 78, 79, 80~~

x=5

1) Midpoint is 50 (13/2=6.5, 7th number is the midpoint)
2) x< midpoint (5<50), ignore the range on the right (50-80)

Review binary search:

Given list of numbers find number 5

1, 2, 5, 7, 8, 10, ~~50, 60, 71, 77, 78, 79, 80~~

x=5

1) Midpoint is 50 (13/2=6.5 7th number is the midpoint)
2) x< midpoint (5<50), ignore the range on the right (50-80)
3) Midpoint is 5

To identify new midpoint:

1. Count the total number of left number entries (which is 6)
2. Divide the total number by 2 (in this example, 6/2=3,
3. 3rd number is the new midpoint

Review binary search:

Given list of numbers find number 5

1, 2, 5, 7, 8, 10, ~~50, 60, 71, 77, 78, 79, 80~~

x=5

1) Midpoint is 50
2) x< midpoint (2<50), ignore the range on the right (50-80)
3) Midpoint is 5
4) x is equal to midpoint (5==5). The number is found!

Task 3

Try to apply binary search on a following set of elements:

1, 2, 5, 6, 8, 10, 50, 70, 75, 77, 78, 79, 80, 81, 82, 85

Find number 77

Try to come up with solution

Task 3 answer

1, 2, 5, 6, 8, 10, 50, 70, 75, 77, 78, 79, 80, 81, 82, 85

x=77 (the number we are searching)

1) Midpoint is 70 (16/2=8, 8th digit)

Task 3 answer

1, 2, 5, 6, 8, 10, 50, 70, 75, 77, 78, 79, 80, 81, 82, 85

x=77 (the number we are searching)

1) Midpoint is 70 (16/2=8, 8th digit)
2) x>midpoint (77>70), ignore the range on the left (1-70)
3) Midpoint is 79 (8/2=4, 4th digit in the remaining set)

Task 3 answer

1, 2, 5, 6, 8, 10, 50, 70, 75, 77, 78, 79, 80, 81, 82, 85

x=77 (the number we are searching)

1) Midpoint is 70 (16/2=8, 8th digit)
2) x>midpoint (77>70), ignore the range on the left (1-70)
3) Midpoint is 79 (8/2=4, 4th digit in the remaining set)
4) x<midpoint (77<79), ignore the range on the right (79-85)

Task 3 answer

1, 2, 5, 6, 8, 10, 50, 70, 75, 77, 78, 79, 80, 81, 82, 85

x=77 (the number we are searching)

1) Midpoint is 70 (16/2=8, 8th digit)
2) x>midpoint (77>70), ignore the range on the left (1-70)
3) Midpoint is 79 (8/2=4, 4th digit in the remaining set)
4) x<midpoint (77<79), ignore the range on the right (79-85)
5) Midpoint is 77(3/2=1.5, 2nd digit)
6) X is equal to midpoint (77==77), number is found

Task 4.1

Try implementing binary search (recursively) in Python.

Adopt the solution provided during the lecture

Push your solution to Github

Solution to Task 4.1

https://github.com/wiut-tutor/seminar10/blob/main/task4.1.py

Solution to Task 4.1

```
1   def binarySearchRecursive(sequence, x, left, right):
2           if left > right:
3               return False
4           mid = (left+right)//2
5           print('mid', sequence[mid])
6
7           if x == sequence[mid]:
8               return mid
9           elif x < sequence[mid]:
10              return binarySearchRecursive(sequence, x, left, mid - 1)
11          elif x > sequence[mid]:
12              return binarySearchRecursive(sequence, x, mid + 1, right)
13
14
15  if __name__ == "__main__":
16      sample = [1, 2, 5, 6, 8, 10, 50, 70, 75, 77, 78, 79, 80, 81, 82, 85]
17      x = 77
18      print('Index of number ', x, 'is equal to ', binarySearchRecursive(sample, x, 0, len(sample) - 1))
```

Task 4.1

Try to update previous code to make binary search work iteratively

Hint: try while loop

Push your solution to Github

Solution to Task 4.2

https://github.com/wiut-tutor/seminar10/blob/main/task4.2.py

Solution to Task 4.2

```python
def binarySearchIterative(sequence, x):
    left = 0
    right = len(sequence)-1
    mid = 0

    while left <= right:
        mid = ((left + right) // 2)
        print('mid', sequence[mid])
        if x == sequence[mid]:
            return mid
        elif x < sequence[mid]:
            right = mid-1

        elif x > sequence[mid]:
            left = mid+1


if __name__ == "__main__":
    sample = [1, 2, 5, 6, 8, 10, 50, 70, 75, 77, 78, 79, 80, 81, 82, 85]
    x = 77
    print('Index of number ', x, 'is equal to ', binarySearchIterative(sample, x))
```

Advanced task: applying binary search

Farmer John has built a new long barn, with N (2 <= N <= 100,000) stalls. The stalls are located along a straight line at positions x1,...,xN (0 <= xi <= 1,000,000,000).

His C (2 <= C <= N) cows don't like this barn layout and become aggressive towards each other once put into a stall. To prevent the cows from hurting each other, FJ wants to assign the cows to the stalls, such that the minimum distance between any two of them is as large as possible. What is the largest minimum distance?

Input

t – the number of test cases, then t test cases follows.

* Line 1: Two space-separated integers: N and C

* Lines 2..N+1: Line i+1 contains an integer stall location, xi

Output

For each test case output one integer: the largest minimum distance.

Example

https://www.includehelp.com/data-structure-tutorial/aggressive-cows-on-binary-search.aspx

Constraints:

```
N: 2<=N<=100,000
C: 2<=C<=N
Xi: 0<=Xi<=1,000,000,000
```

Input:

```
Line1: Two integers N and C
Next N lines: An integer for Xi stall's position
```

Example:

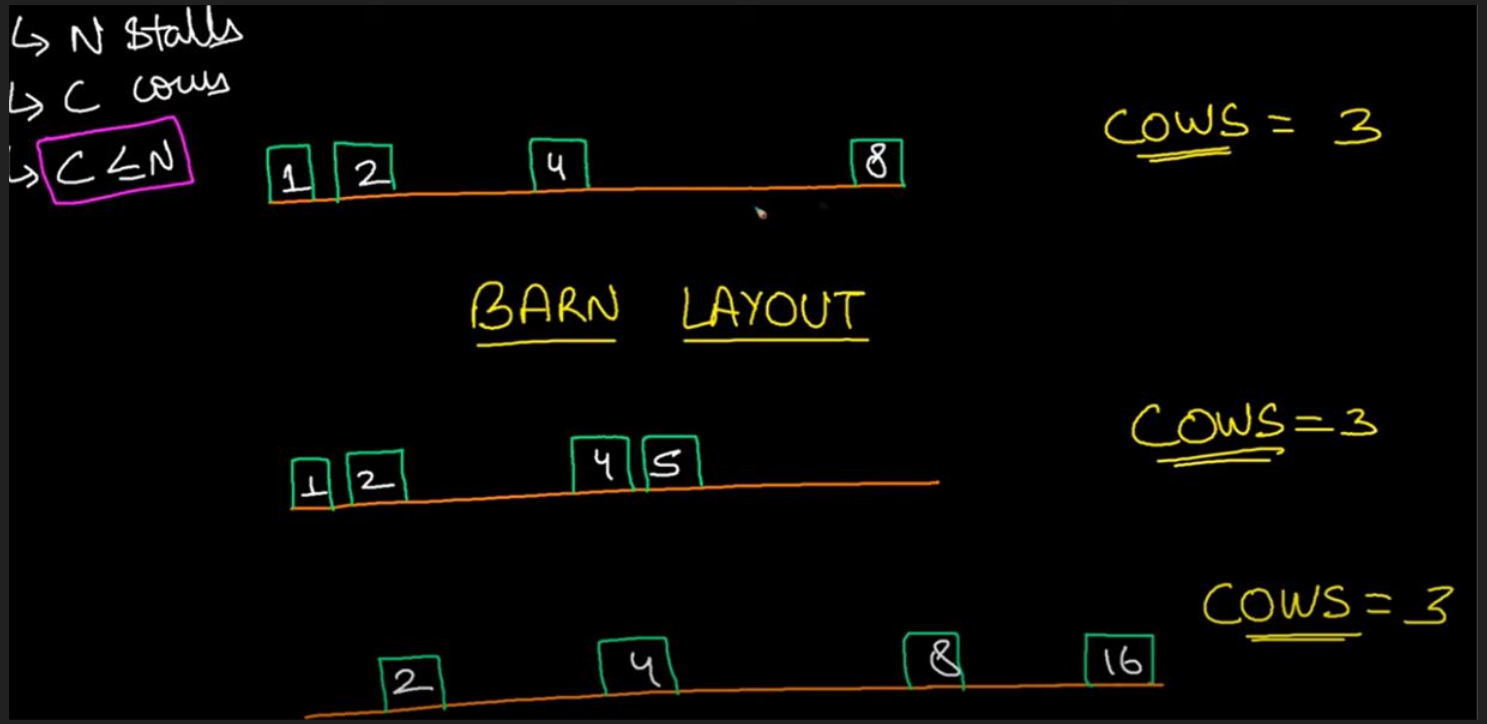
```
Input:
5 3
1 2 4 8 9

Output:
3
```

Explanation:

If the cows are placed at 1,4 and 9, it will result in minimum distance from 3.

Example

https://www.youtube.com/watch?v=SiE1XFhYoaA



Recommended reading

Computer Science Illuminated, Chapter 7-9, end of the book exercises

https://adrianmejia.com/most-popular-algorithms-time-complexity-every-programmer-should-know-free-online-tutorial-course/

Explore:

http://carlcheo.com/startcoding

Recommended sources

- Computer Science Illuminated, Chapter 7-9, end of the book exercises
- Intro to Algorithms - Crash course
- https://introprogramming.info/english-intro-csharp-book/read-online/chapter-19-data-structures-and-algorithm-complexity/
- https://www.khanacademy.org/computing/computer-science/algorithms/binary-search/a/binary-search
- https://adrianmejia.com/how-you-can-change-the-world-learning-data-structures-algorithms-free-online-course-tutorial/
- https://www.techrepublic.com/article/why-clean-code-is-more-important-than-efficient-code/

References

1. Dale, N., & Lewis, J. (2020). Computer Science Illuminated (7th ed.). Jones & Bartlett Learning, Chapter 7-9
2. “Intro to Algorithms: Crash Course Computer Science #13.” YouTube, 24 May 2017, www.youtube.com/watch?v=rL8X2mlNHPM.
3. Svetlin Nakov. “Introduction to Programming with C# / Java Books» Chapter 19. Data Structures and Algorithm Complexity.” Introprogramming.info, 2019, introprogramming.info/english-intro-csharp-book/read-online/chapter-19-data-structures-and-algorithm-complexity/.
4. Mejia, Adrian. “How You Can Change the World by Learning Data Structures and Algorithms.” Adrian Mejia Blog, adrianmejia.com/how-you-can-change-the-world-learning-data-structures-algorithms-free-online-course-tutorial/.
5. TECH DOSE. “Aggressive Cow | SPOJ.” Www.youtube.com, www.youtube.com/watch?v=SiE1XFhYoaA. Accessed 9 June 2022.
6. apotheon. “Why Clean Code Is More Important than Efficient Code.” TechRepublic, 10 June 2011, www.techrepublic.com/article/why-clean-code-is-more-important-than-efficient-code/. Accessed 9 June 2022.
7. “Aggressive Cows (on Binary Search).” Www.includehelp.com, www.includehelp.com/data-structure-tutorial/aggressive-cows-on-binary-search.aspx. Accessed 9 June 2022.

Thank you