

Module - 2

(Second year)

Binary Search

Resources

- <https://www.topcoder.com/community/competitive-programming/tutorials/binary-search>
- https://en.wikipedia.org/wiki/Binary_search_algorithm
- Complexity: <https://www.geeksforgeeks.org/complexity-analysis-of-binary-search>
- Lower bound: http://www.cplusplus.com/reference/algorithm/lower_bound/
- Upper bound: http://www.cplusplus.com/reference/algorithm/upper_bound/

Problems

- <https://www.interviewbit.com/courses/programming/topics/binary-search/#problems>
- [Shared in session](#)

Priority queues

Resources

- https://en.wikipedia.org/wiki/Priority_queue
- <https://www.cs.cmu.edu/~wlovas/15122-r11/lectures/15-priorqs.pdf>
- <https://www.geeksforgeeks.org/binary-heap/>
- C++: <https://www.geeksforgeeks.org/implement-min-heap-using-stl/>
- Java: <https://www.geeksforgeeks.org/priority-queue-class-in-java-2/>
- Python: <https://www.geeksforgeeks.org/heap-queue-or-heapq-in-python/>

Problems

- [IPCTRAIN](#), [editorial](#)
- [ANUMLA](#), [editorial](#)
- [KSUBSUM](#), [editorial](#)
- [RRATING](#), [editorial](#)
- [TSECJ05](#), [editorial](#)
- [WEIRDFN](#)
- [CAPIMOVE](#), [editorial](#)
- [RMID2](#)

- [LAZYPROG](#)
- [EXPEDI](#)
- [Sequence Median](#)
- [Maze Checking and Visualization](#)
- [MOSTDIST](#), [editorial](#)

Note

You can skip how a heap is made (using a tree). Still the resources regarding the same have been provided for the interested people. You just need to the basics and how to use priority queue.

Linear Recurrences

Resources

- fusharblog.com/solving-linear-recurrence-for-programming-contest/
- <https://www.geeksforgeeks.org/matrix-exponentiation/>
- <https://www.topcoder.com/tc?module=Static&d1=features&d2=010408>
- <http://zobayer.blogspot.com/2010/11/matrix-exponentiation.html>

Problems

- <https://discuss.codechef.com/tags/recurrence/>
- <https://projecteuler.net/problem=258>
- [SPOJ- MATEX](#)

Data Structures

You are expected to learn basic data structures like stacks, queues, linked lists and trees on your own in a month. The resources regarding the same are easy to obtain and thus are not included in the contents of the module. For trees, you can skip the related algorithms for now. If you need further guidance regarding the same, ping us at t.me/codeshows.