CIT 596

Binary Search and Introduction Randomized Algorithms

SEARCHING IN AN ARRAY

How long does it take to search for an element x in an array?

$O(n)$

Can we do better if we know that the array is sorted?

BINARY SEARCH

Idea: Comparing x to middle element of array eliminates half the array.

- Each step of the algorithm, the size of the input (the search space) halves.

- $T(n) = T\left(\frac{n}{2}\right) + 1$

- Apply Master Theorem: $a = 1, b = 2$,

- Note that $n^0 = 1.$ So $c = 0$ and $a = b^c$.

- All levels contribute equally and each contributes 1.

- Running time is $O(\log n)$.

BINARY SEARCH – ALGORITHM

```
BinarySearch(A, val, lo, hi):
    if hi < lo
        return −1
    mid ← (lo + hi) / 2
    if A[mid] > val:
        return BinarySearch(A, val, lo, mid − 1)
    else if A[mid] < val:
        return BinarySearch(A, val, mid + 1, hi)
    else
        return mid


# to initially call the function:
BinarySearch(A, val, 0, len(A) − 1)
```

NEXT EXAMPLE

Quicksort: an algorithm that tosses coins!

What does an algorithm gain by tossing coins?

- Remember we are interested in the worst-case behavior of an algorithm.

- A deterministic algorithm (one with no randomization) can have a bad worst-case behavior.

- Think about a penalty kicker in soccer as an algorithm that can shoot right or left. If it always does the same thing, the goalie can easily anticipate it and block the shot.

- A worst-case goalie will cause the kicker to have no success! Randomizing helps.

WHY RANDOMIZE?

- In the previous example, the algorithm is like the kicker, and the goalie is like an adversary (opponent) trying to make the algorithm look bad.

- Similarly, think of an adversary creating the inputs for an algorithm based on how it works.

  - Of course, there is no adversary, but since we want to consider worst-case performance, we can imagine the worst case being constructed by an adversary (like the goalie).

- Randomizing helps to avoid such worst cases.

EXAMPLE: RANDOMIZATION IN ROCK-PAPER-SCISSORS

Assignment Project Exam Help

https://tutorcs.com

WeChat: cstutorcs