Algorithms and Time Complexity Chapter 7.10

Course: CPSC 1150

Instructor: Dr. Bita Shadgar

Lecture 20

Learning Outcomes

- Choose the efficient algorithm for solving a problem
- Define a metric to measure algorithms
- Design and implement a search algorithm
  - Linear search
  - Binary search

Opening problem

New web browsers use 128 bits encryption code to secure critical data.

– How long does it take to break the code by checking all different combinations? Having a computer that can test a billion combinations per second.

A. About five minutes.
B. Less than 5 hours.
C. It may take more than 5 hours, but it will definitely take less than 5 days.
D. It may take 50 days.
E. None of them.

Solution

- 2^{128} different combinations.
- $2^{128} = 3.4 \times 10^{38}$
- Every day is equal to 3600 * 24 = 86,400 seconds.
- The computer is able to check 86400 * 1,000,000,000 = 8.64×10^{13} codes per day.
- As a result, it takes $3.4 \times 10^{38} / 8.64 \times 10^{13} = 39.3 \times 10^{23}$ days
- Or $39.3 \times 10^{23} / 365 = 10.7 \times 10^{21}$ years
- Remember that the universe is about 15×10^{9} years old, and

$$15 \times 10^{9} \ll 10.7 \times 10^{21}$$

Time Complexity

- Why it takes so much time?
  - Because the complexity of this algorithm is exponential (2^n)
- Consider a program to sum all elements of an array:

```
int sum=0;
for(int i=0; i< arr.length; i++)
      sum+=arr[i];
```

- Assume array contains 50 elements. If this program takes 50 time units to execute and sum all elements of this array, then if array contains 100 elements, it may take 100 time units.
- So, the time needed for this program is proportional to n, length of array.
- The complexity of this algorithm is linear, O(n)

Searching

- Process of finding a target or key in a list is called Searching
- Linear (Sequential) search
  - Search **35** in the following list – how many comparisons do you need?

13	**7**	**12**	**35**	**40**	**53**	**23**	**9**	**74**	**6**

  - Search **45** in the list -

Binary Search

– Search **23** in the following list – how many comparisons do you need?

0	1	2	3	4	5	6	7	8	9
6	7	9	12	13	23	35	40	53	74

Start= 0 mid = 4 end = 9

5	6	7	8	9
23	35	40	53	74

Start= 5 mid = 7 end = 9

5	6
23	35

Start=mid= 5 end = 6

Implementing Binary Search

– Write a method to take an array of integers and an integer value as target. The method should search the array and returns the index of target found in the array. If there is no such a number, the method should return -1.