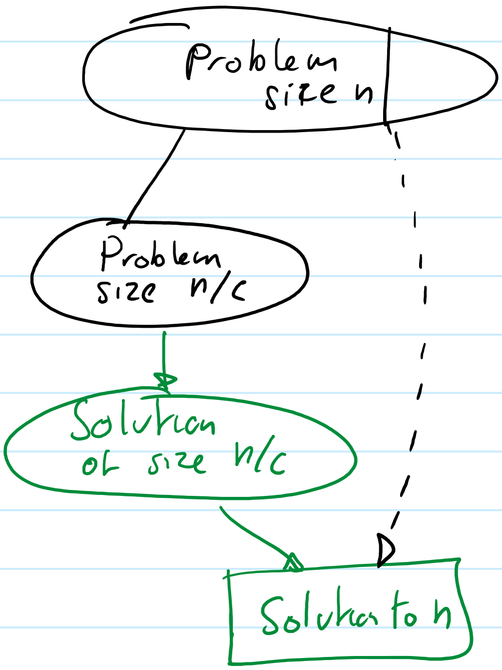
6.2 Decrease-by-a-Constant-Factor Algorithms

Tuesday, October 22, 2024 11:05 AM

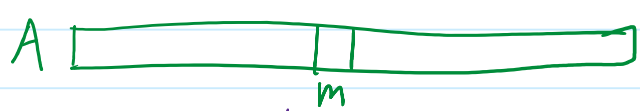


- Example Algorithm: Binary-Search

Problem: Given a sorted array A[0 ... n-1]
answer whether a value k is in A

idea: How to find a solution for size = n
from the solution to size = n/c

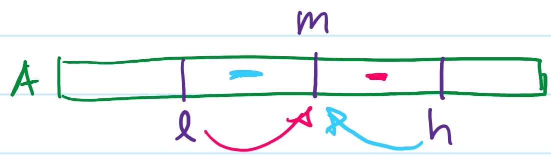
Algorithm:
compare k to the element in the middle of the array



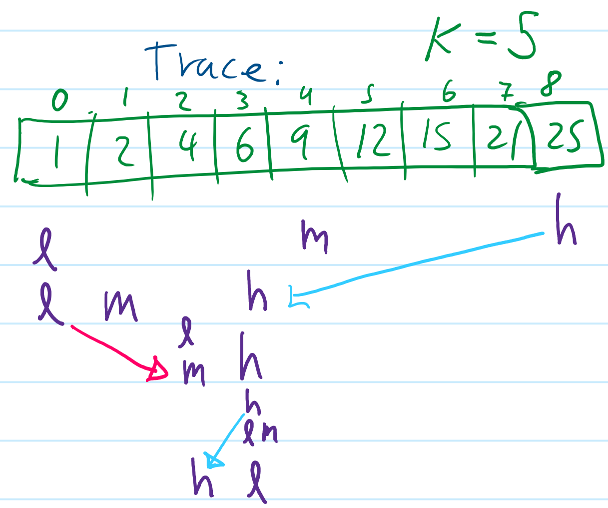
cases
- $A[m] = k$ success!!
- $k < A[m]$: search in the left subarray
- $k > A[m]$: search in the right subarray.

- Solution can be recursive:
  but final implementation can be iterative.

• Iterative version.




```
FUNCTION BinarySearch(A[0..n-1], K)
  l ← 0
  h ← n-1
  while l ≤ h do
      m ← ⌊(l+h)/2⌋
      if K = A[m] then
          return m
      elsif K < A[m] then
          h ← m-1
      else
          l ← m+1
  return -1
```



- Analysis: basic operation.

$$C_{worst}(n) = 1 + C(n/2)$$

$$C_{best}(n) = 1$$

$$C(1) = 1$$

⇓ (many steps)

$$C(n) = \log_2 n + 1$$

$$C(n) \in \Theta(\log n)$$

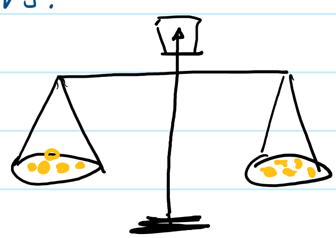
• key points:
- Finding elements is extremely useful
- It's good that we can do it fast

- Finding elements is extremely useful
- It's good that we can do it fast

n	Linear Search	Binary Search
30,000	30,000	15
3,000,000	3,000,000	22
300,000,000	300,000,000	29

- Example: Fake-Coin Problem

Suppose you have n identical coins
- one coin is a fake
- the fake coin has a different weight than the real ones, it is lighter.
- you only have a balance to compare groups of coins.



find the fake coin!

idea: How to find a solution for size $= n$ from the solution to size $= n/c$

- Split into two piles:

case n is even $\left(\frac{n}{2}\right)$ vs $\left(\frac{n}{2}\right)$
- continue search on pile that is lighter.

case n is odd: $\left(\lfloor\frac{n}{2}\rfloor\right)$ vs $\left(\lfloor\frac{n}{2}\rfloor\right)$, 1

- continue seach on pile that is lighter.
- if piles balance, the single coin is the fake one.

Analysis. basic_operation weighting

$$W(n) = 1 + W\left(\frac{n}{2}\right)$$
$$W(1) = 0$$

$\Downarrow$ in many steps

$$W(n) = \log_2 n \in \Theta(\log n)$$

Can we do better?

split by 3

$$\left(\frac{n}{3}\right) \text{ vs } \left(\frac{n}{3}\right), \quad \left(\frac{n}{3}\right)$$

- continue search on pile that is lighter.
- if piles balance, continue seach in left over pile.

remainders: 0:

1: live reminder with outside pile

2: Split coins over weighted piles.

- Example Algorithm: Russian Peasant Multiplication

$$n \cdot m = ?$$

how to find a solution for n that depends how to find a solution for n that depends on $n/2$

- $n \cdot m = \frac{n}{2} \cdot 2m$ if n is even.
- $n \cdot m = \frac{n-1}{2} \cdot 2m + m$ if n is odd.

- multiplying by 2 is easy
- dividing by 2 is easy

$50 * 65$

n	m
50	65
25	130
12	260
6	520
3	1040
1	2080

$$\begin{array}{r} 130 \\ 1040 \\ 2080 \\ \hline 3250 \end{array}$$

$$\begin{array}{r} 50 \\ * \ 65 \\ \hline \end{array}$$

Note: multiplying and dividing by 2 are very cheap in binary.