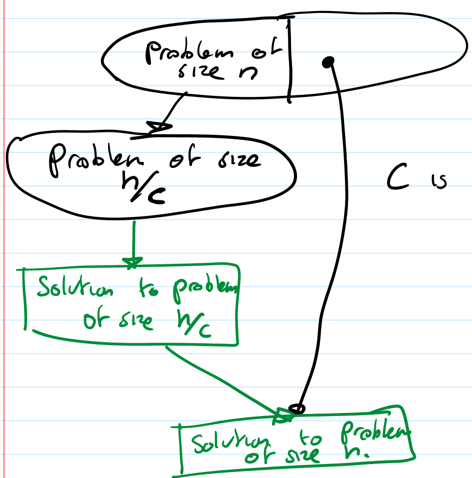
6.2 Decrease-by-a-Constant-Factor Algorithms

Tuesday, October 17, 2023 11:19 AM



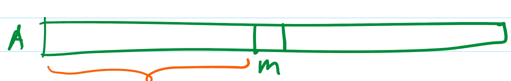
C is usually 2.

- Algorithm #1 Binary-Search.

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

Draft Solution.
How to find solution to size = n from solution to size = $\frac{n}{2}$

Algorithm Draft:
- compare K to element in the middle of A $A[m]$

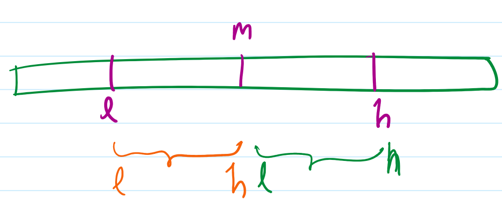


- if $K = A[m]$ return true
- if $K < A[m]$ ask for search in lower half
- if $K > A[m]$ ask for search in upper half.

- Solution can be designed recursively. But final implementation can be iterative.

- Iterative version: need l low, h high > the portion of the array being searched.

Algorithm:




```
FUNCTION  BinarySearch(A[0..n-1], K)
    l ← 0
    h ← n-1
    WHILE l ≤ h DO
        m = ⌊(l+h)/2⌋
        IF K = A[m]
            return m
        ELSIF K < A[m]
            h ← m-1
```

return m
ELSIF $k < A[m]$
$h \leftarrow m-1$
ELSE
$l \leftarrow m+1$
RETURN -1

- Analysis:
  basic operator =

$\cdot\ C(n) = 1 + C(\frac{n}{2})$
$\cdot\ C(1) = 1$
$\Downarrow$
$\cdot\ C(n) = \log n + 1$
$C(n) \in \Theta(\log n)$.

Let $n = 2^k$ $\quad \underline{k = \log n}$

$\cdot\ C(2^k) = \underbrace{C(2^{k-1})} + 1$

by backwards subst.
$\bullet\ C(2^{k-1}) = C(2^{k-2}) + 1$

$C(2^k) = C(2^{k-2}) + 1 + 1$

$\cdot\ C(2^{k-2}) = C(2^{k-3}) + 1$

$C(2^k) = C(2^{k-3}) + 1 + 1 + 1$

after i substitutions

$C(2^k) = C(2^{k-i}) + i$

Let $i = k$

$C(2^k) = C(2^{k-k}) + k$

$C(2^k) = 1 + k$

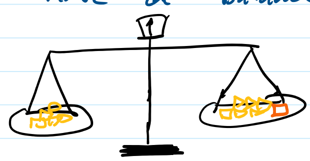
Key points:

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

n	Binary Search	Linear Search
30,000	15	30,000
3,000,000	22	3,000,000
300,000,000	28	300,000,000

Problem #2 Fake-Coin problem.

Suppose you have a pile of identical coins.
- One coin is fake.
- The fake coin does not weight the same as the real coins, it is lighter.
- You only have a balance to compare coins.



Kevins solution
- Split the coins in two, weight them.

How to solve size $= n$ from solution to size $= n/2$

n is even: $(\frac{n}{2})$ vs $(n/2)$

n is even: $\left(\frac{n}{2}\right)$ vs $\left(n/2\right)$

continue search on pile that is lighter

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

- continue search on pile that is lighter
- if plates are balanced, the single coin is the fake one.

Analysis:

basic operation: weighting.

$$W(n) = 1 + W\left(\frac{n}{2}\right)$$
$$W(1) = 0$$
$$\Downarrow$$
$$W(n) = \log_2 n \in \Theta(\log_2 n)$$

Can you do Better?

Yes: By splitting into 3 piles.

Algorithm: Russian peasant multiplication.

$$n \cdot m = ?$$

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

$$n \cdot m = \frac{n}{2} \cdot 2m \quad \text{if } n \text{ is even.}$$

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

but, only by 2 .- multiplying by 2 is easy
halving by 2 is easy.

cont....