CSC236 fall 2018

recursive time complexity

difficult road to laziness...
...we'll get there next week...

Danny Heap

heap@cs.toronto.edu / BA4270 (behind elevators)

http://www.teach.cs.toronto.edu/~heap/236/F18/

416-978-5899

Using Introduction to the Theory of Computation,
Chapter 3
binary search

Recursive $T(n)$

correctness.... later
worst-case time complexity

$x$: value to be searched for
$A$: array
$b$: beginning index
$e$: end index

$n = \text{len}(A[b:e+1]) = e-b+1$

def recBinSearch(x, A, b, e):
    if $b == e$:
        if $x \leq A[b]$:
            return $b$
        else:
            return $e + 1$
    else:
        $m = (b + e) // 2$  # midpoint
        if $x \leq A[m]$:
            return recBinSearch(x, A, b, m)
        else:
            return recBinSearch(x, A, $m+1$, e)

exercise to reader: show that $m-b+1 = \text{ceiling}(n/2)$, and $e-(m+1)+1 = \text{floor}(n/2)$
show that $T$ is nondecreasing

$T(n) = \begin{cases} 
  1 & \text{if } n = 1 \\
  1 + \max(T(\text{ceiling}(n/2)), T(\text{floor}(n/2))) & \text{otherwise}
\end{cases}$

c'' ..... = 1 (choosing units of c'').
suppose \( n = 2^k \), for some natural number \( k \) (bigger than 0, for now).

\[
T(n) = T(2^k) = 1 + T(2^{k-1})
= 1 + 1 + T(2^{k-2}) = 2 + T(2^{k-2})
= 3 + T(2^{k-3})
\]

... intuition happens here!

\[
= k + T(2^{k-k}) = \lg(n) + c'
\]

conjecture: \( T \in \Theta(\lg) \)

want to prove \( T \in \Omega(\lg) \) [then big-Oh later...]
prove lower bound on $T(n)$

Let $c = ???$. Then $c \in \mathbb{R}^+$. Let $B = ???$. Then $B \in \mathbb{R}^+$.

(complete induction)

Let $n$ be an arbitrary natural number no smaller than $B$. Assume \( \forall B \leq i < n, T(i) \geq c \log(i) \). I will show that $T(n) \geq c \log(n)$.

**case** $n \geq 3$: $T(n) = 1 + T(\lceil n/2 \rceil)$ \# since $n \geq B > 1$

$\geq 1 + c \log(\lceil n/2 \rceil)$ \# by IH, since $B \leq \lceil n/2 \rceil < n$, since $n \geq 3$

$\geq 1 + c \log(n/2)$ \# since $\log$ nondecreasing

$= 1 + c(\log(n) - \log(2)) = 1 - c + c \log(n)$

$\geq c \log(n)$ \# since $c \leq 1$

**base case** $n = 2$: Then $T(2) = 1 + T(1) = 1 + c' \geq c \log(2) = c$ \# since $c = 1$
try to prove upper bound on \( T(n) \)

trouble!?!?

Let \( c = ??? \). Then \( c \in \mathbb{R}^+ \). Let \( B = n??? \). Then \( B \in \mathbb{R}^+ \).

(complete induction)

Let \( n \) be an arbitrary natural number no smaller than \( B \). Assume (IH) \( \forall B \leq i < n, T(i) \leq c \lg(i) \). I will try to show that \( T(n) \leq c \lg(n) \).

case \( n \geq B \): \( T(n) = 1 + T(\text{ceiling}(n/2)) \) \( \# \) since \( n \geq B > 1 \)
\( \leq 1 + c \lg(\text{ceiling}(n/2)) \) \# by IH, since \( B \leq \text{ceiling}(n/2) < n, \) since \( n > 2 \)
\( \leq 1 + c \lg((n+1)/2) \) \# since \( \lg \) is nondecreasing
\( = 1 + c(\lg(n+1) - 1) = 1 - c + c \lg(n+1) \)
\( \leq c \lg(n) \)........................darn!

strengthen the claim: \( T(n) \leq c \lg(n-1) \)

\begin{align*}
\text{case } n \geq B: \quad T(n) &= 1 + T(\text{ceiling}(n/2)) \quad \# \text{ since } n \geq B > 1 \\
&\leq 1 + c \lg(\text{ceiling}(n/2)-1) \quad \# \text{ by IH, since } B \leq \text{ceiling}(n/2) < n, \text{ since } n > 2 \\
&\leq 1 + c \lg((n+1)/2 -1) \quad \# \text{ since } \lg \text{ is nondecreasing} \\
&= 1 + c \lg((n+1-2)/2) = 1 + c \lg((n-1)/2) = 1 + c(\lg(n-1) - 1) \\
&= 1 - c + c \lg(n-1) \\
&\leq c \lg(n-1) \quad \# \text{ since } c \geq 1
\end{align*}

\begin{align*}
\text{case } n = 2: \quad T(2) &= 1 + c' \leq c \lg(1) \ldots \ldots \ldots \text{won't work} \\
\text{case } n = 3: \quad T(3) &= 1 + T(2) = 2 + c' \leq c \lg(2) = c \quad \# \text{ true since } c = 2+c'
\end{align*}
Notes