

CSC236 fall 2016

time complexity

Danny Heap

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

<http://www.cdf.toronto.edu/~csc236h/fall/>

416-978-5899

Using Introduction to the Theory of Computation,
Chapter 3

Outline

complexity of recursive functions

Notes

binary search

Recursive $T(n)$

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



Lower bound on $T(n)$... by unwinding

Lower bound on $T(n)$... by proof



Upper bound on $T(n)$

trouble!



Notes