Lecture 5, Oct. 6, 2009

Total: $\Theta(n^{\lg_b a}) + \sum_{i=0}^{\lg_b n-1} a^i f(n/b^i)$

Third case

$a^j f(n/b^j) \le c^j f(n)$ for some $c < 1$, and $f(n) = \Omega(n^{\lg_b a+\varepsilon})$

$$\Rightarrow \sum_{i=0}^{\lg_b n-1} c^i f(n) = \Theta(f(n))$$

$$\Rightarrow \sum_{i=0}^{\lg_b n-1} a^i f(n/b^i) = O(f(n))$$ Note Big-Oh and not Theta !

The first term is $\Theta(n^{\lg_b a}) = O(f(n))$

TOTAL: $\Theta(f(n))$

Why Theta and not plain big-O ?

48

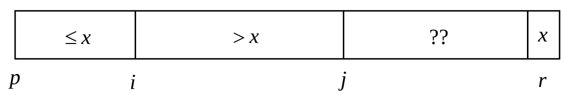
Back to algorithms - Quicksort

- Quicksort
  - » Sort in place
  - » Practical
  - » Divide & Conquer
- Algorithm:
  - » Divide into 2 arrays around the first element
  - » recursively sort each array
  - » merge/combine - trivial.

56

Partition routine

- ```
  Partition(A,p,r)
    x=A(r)
    i=p-1
    for j=p to r-1
      if A(j) <= x
        then i++, exchange A(i) and A(j)
    exchange A(i+1) and A(r)
    return (i+1)
  ```



Partition - Example

7 8 3 5 4 1 9 6

i *j* *x*

7 8 3 5 4 1 9 6

i *j* *x*

7 8 3 5 4 1 9 6 -- now exchange 3 and 7, advance i.

i *j* *x*

3 8 7 5 4 1 9 6

i *j* *x*
  ```

Quicksort

- ```
  Quicksort(A,p,r)
    if p<r
      q=partition(A,p,r)
      quicksort(A,p,q-1)
      quicksort(A,q+1,r)
  ```

- To simplify, assume distinct elements.
  - Lucky - always an even split: $T(n)=2T(n/2)+\Theta(n) \Rightarrow T(n)=\Theta(n\lg n)$
  - Unlucky: $T(n)=T(0)+T(n-1)+\Theta(n) \Rightarrow \mathbf{T(n)=\Theta(n^2)}$

- How to avoid bad case ?
  - Partitioning around middle element does not work!
  - Idea: partition around a random element.

Randomized Algorithms

- Algorithm can "toss coins".
- No specific input leads to worst-case behavior.
- Distinction between randomized algorithms and random data !
  ```

Analyzing Quicksort

- Partition around a randomly chosen element and let T(n) be the expected time to sort.
- Consider the case where the partition is (k, n-k-1). Conditioned on partition turning to be (k, n-k-1), the expected time to terminate is:

$$T(k)+T(n-1-k)+\Theta(n)$$

- Note that any value of k, from 0 to n-1 is equally likely.

$$T(n) = E_k\left[T(n \mid (k, n-k-1) \text{ split})\right]$$
$$= \sum_k \Pr[(k, n-k-1) \text{ split}]\, T(n \mid (k, n-k-1) \text{ split})$$
$$= \frac{1}{n}\sum_k \left[T(k)+T(n-1-k)+\Theta(n)\right]$$
$$= \frac{2}{n}\sum_{k=0}^{n-1}\left[T(k)\right]+\Theta(n)$$

61

Solving the recurrence

We will try to prove that $T(n) \le an\lg n + b$

First, choose b large enough to satisfy: $T(1) \le a\lg 1 + b = b$

Inductive step (note that $T(0)=0$):

$$T(n) = \frac{2}{n}\sum_{k=0}^{n-1} T(k) + \Theta(n) \le \frac{2}{n}\sum_{k=0}^{n-1}(ak\lg k + b) + \Theta(n)$$

$$= \frac{2}{n} a \underbrace{\sum_{k=1}^{n-1} k\lg k} + \frac{2}{n} nb + \Theta(n)$$

Need to prove that this is $\le \frac{1}{2}n^2\lg n - \frac{1}{8}n^2$

Note that using $\sum_{k=1}^{n-1} k\lg k \le n^2 \lg n$ is not enough !!

$$\le \frac{2}{n} a\left(\frac{1}{2}n^2\lg n - \frac{1}{8}n^2\right) + 2b + \Theta(n)$$

$$= an\lg n + b + \underbrace{\left(\Theta(n) + b - an/4\right)}_{\le 0 \text{ for large enough } a}$$

62

Technical lemma

$n^2 \lg n$ bound is trivial. Need a stronger bound

$$\sum_{k=1}^{n-1} k \lg k = \sum_{k=1}^{\lceil \frac{n}{2}-1 \rceil} k \lg k + \sum_{\lceil \frac{n}{2} \rceil}^{n-1} k \lg k$$

$\le \lg(n/2) = \lg n - 1$

$$\le \sum_{k=1}^{\lceil \frac{n}{2}-1 \rceil} k \lg n - \sum_{k=1}^{\lceil \frac{n}{2}-1 \rceil} k + \sum_{\lceil \frac{n}{2} \rceil}^{n-1} k \lg n$$

$$\le \lg n \sum_{k=1}^{n-1} k - \sum_{k=1}^{\lceil \frac{n}{2}-1 \rceil} k$$

$$\le \lg n \frac{n(n-1)}{2} - \frac{(n/2-1)(n/2)}{2}$$

$$\le \frac{1}{2} n^2 \lg n - \frac{n^2}{8}$$

HW: We proved O, now prove Ω.

63