3137 Data Structures and Algorithms in C++

Lecture 7 July 26 2006 Shlomo Hershkop

1

2

Announcements

will do review later today

will take questions at end

please make sure to submit/plan hw

semester is going to end in 2 weeks from today

Outline

Sorting – quick sort
Disjoint DS
Review for midterm

Reading: Chapter 7.7-7.8, 8-8.3













P1V0t #2

choose random element for pivot

□ pro/cons ?

<text><text>

Pivot	#3
	ΠJ

Choose median value from the list

□ pro/cons ?

hmmm don't you need a sorted list to get median?

actually there is a linear algorithm for this
 will be doing it on homework

11



coding

ok so enough theory, how do you code all this ??

arrays are much cheaper than linked lists

Iots of tricks to keep things cheap

```
1 /**
2 * Quicksort algorithm (driver).
3 */
4 template <typename Comparable>
5 void quicksort( vector<Comparable> & a )
6 {
7 quicksort( a, 0, a.size( ) - 1 );
8 }
```

```
15
```

```
/**
 1
 2
      * Return median of left, center, and right.
 3
      * Order these and hide the pivot.
 4
      */
 5
     template <typename Comparable>
 6
     const Comparable & median3( vector<Comparable> & a, int left, int right )
 7
     {
 8
         int center = ( left + right ) / 2;
 9
         if( a[ center ] < a[ left ] )</pre>
10
             swap( a[ left ], a[ center ] );
         if( a[ right ] < a[ left ] )</pre>
11
12
             swap( a[ left ], a[ right ] );
13
         if( a[ right ] < a[ center ] )</pre>
14
             swap( a[ center ], a[ right ] );
15
16
             // Place pivot at position right - 1
17
         swap( a[ center ], a[ right - 1 ] );
18
         return a[ right - 1 ];
19
    }
                                                                             ...
```





Analysis

so how to analyze quick sort

think how we did mergesort analysis

Quick sort

i = size of left partition
C₁ = time to choose pivot
C₂ = partitioning the set

what do you get?

20





<text><text><text><equation-block><equation-block><equation-block><equation-block><text>

Bottom line

$$\frac{T(N)}{N+1} = \frac{T(1)}{2} + 2C\sum_{i=0}^{N+1} \frac{1}{i}$$

simplifies
$$\frac{T(N)}{N+1} = O(\log N)$$
$$T(N) = N \log N$$

















a log(N!) = log (N * N-1 * N-2 * N-3 log of product = log of sums)
b log(N!) = log(N) + log(N-1) + log(N-2)...
b (d rop n/2 terms) ≥ log(N) + ... + log(N/2)
b n/2 log(N/2) = ON log N)





























linked lists

each member will be in its own list

merge ?find ?

• will come back to this next week











Lists	
arrays	
Iinked lists	
	57









- complete trees
- binary trees
- BST
- Balanced BST
 - AVL
 - Red-black
- Iazy deletion
- tree traversal algorithms
- expression trees
- □ B+ trees
- huffman trees

<text><text><text><text>

















Next

- do all the reading
- the exam is open brain/notes/book closed general internet
- will post when test is ready
- please email/aim but will only answer up to test time
- Good luck