

Computers in Engineering COMP 208

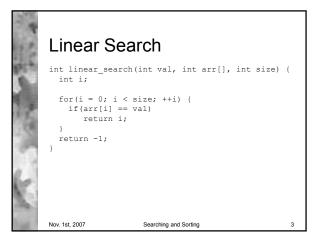
Searching and Sorting Michael A. Hawker

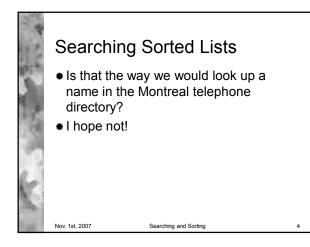
Where's Waldo?

Nov. 1st, 2007

- A common use for computers is to search for the whereabouts of a specific item in a list
- The most straightforward approach is just to start looking at the beginning and go on from there

Searching and Sorting





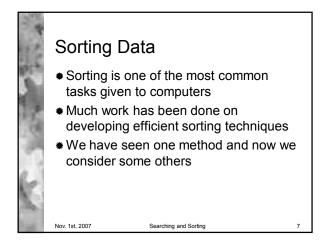
Binary Search

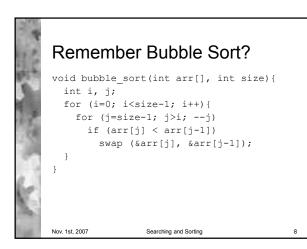
Nov. 1st, 2007

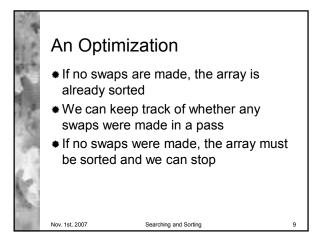
- To search a sorted array, we could check the middle element
- The value we are looking for might be there
- If not we can determine whether it is in the first or second half of the array and search that smaller array

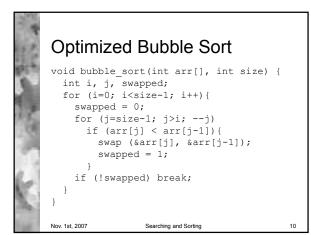
Searching and Sorting













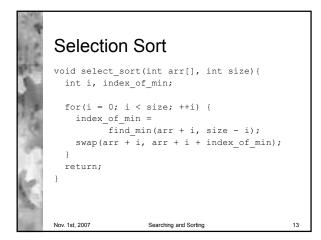
Selection Sort Another sorting technique is known as selection sort At each step, select the smallest value not yet in place and put it where it belongs Where's that? After the smaller elements at the front of the array

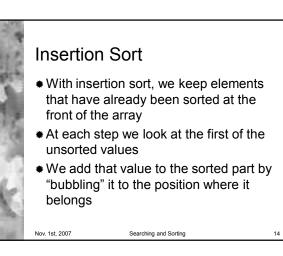
Selection Sort

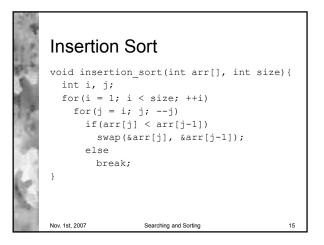
Nov. 1st, 2007

- In the following program, note the use of pointer arithmetic to access the array elements
- * We use arr + i instead of arr[i]
- As an argument arr + i represents an array with starting address arr[i]

Searching and Sorting







The Cost of Algorithms

- * We've seen multiple sorting algorithms
- * Why is one better than the other?
- * How can we measure this?
- In a uniform way?

Finding the Maximum

Nov. 1st, 2007

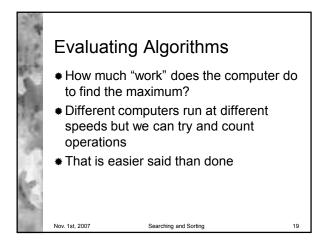
Nov. 1st, 2007

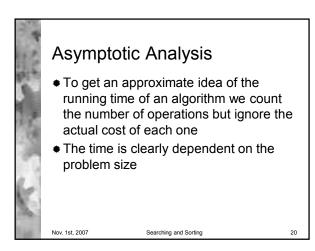
- We have already seen how to find the largest value in an array
- * Here is the C code for that algorithm
- This code returns the location of the largest value (rather than the value itself)

Searching and Sorting

```
Finding Max
int find_max(int arr[], int size) {
    int i, index_of_max = 0;
    for(i = 1; i < size; ++i)
        if(arr[i] > arr[index_of_max])
        index_of_max = i;
    return index_of_max;
}
```

Searching and Sorting





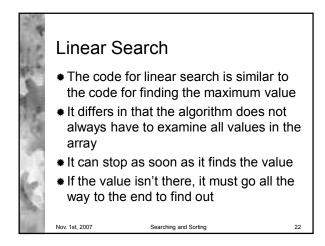
The Cost of Find_Max

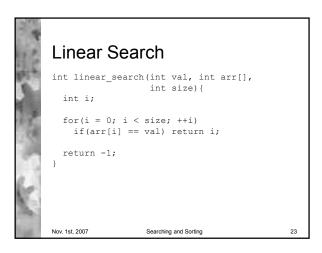
- There is a loop that is executed n-1 times
- Each time a constant number of operations is done

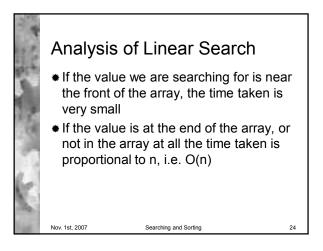
Nov. 1st, 2007

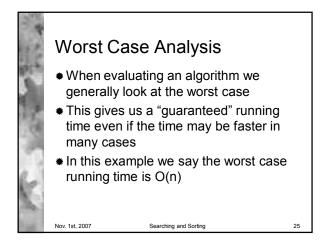
 We say the algorithm for finding the maximum value runs in O(n) time if the problem is of size n

Searching and Sorting









Average Case Analysis

- In general it is difficult to determine the average time an algorithm will take
- Average case time is dependent on the distribution of the data values
- If the data is uniformly distributed and we search for a random value, the average case time for linear search is also O(n)

Searching and Sorting

Binary Search

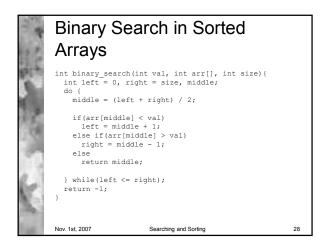
Nov. 1st, 2007

- We have also seen another algorithm for searching sorted lists, binary search
- * Intuitively it seems to be much faster

Searching and Sorting

- How can we show this analytically?
- How much faster is it?

27



The Cost of Binary Search * The original list being searched had n values * After checking the middle element we

- either find the value we are looking for or we reduce the problem size to n/2
- In the worst case, if we don't happen to find the value, the problem size becomes n/4, n/8, n/16, ...

Searching and Sorting

Nov. 1st, 2007

Nov. 1st, 2007

The Cost of Binary Search

- * The process cannot continue forever
- Eventually n/2ⁱ becomes smaller than 1 and the value was either found or is not in the list
- This must stop after log₂ n steps
- The cost of binary search is then O(log n)

Searching and Sorting

30

