

run-time analysis of sorting algorithms

- blort sort cannot compute, since the number of passes made is not predictable in the *best case*, only one pass through the array is made, in the case where the array is in sorted order to begin with. in the *worst case*, the number of passes is infinite...
- selection sort order $N^2 = O(N^2)$ because there is one pass made for each element in the array, i.e., as each element is shifted from the array to be sorted into the auxiliary array, and for each pass, the algorithm looks through the array to find the smallest element to select (which takes O(N))
- insertion sort order $N^2 = O(N^2)$

because there is one pass made for each element in the array, i.e., as each element is shifted from the array to be sorted into the auxiliary array (same as selection sort), and for each pass, the algorithm looks through the auxiliary array to find a position for the new element (which takes O(N))

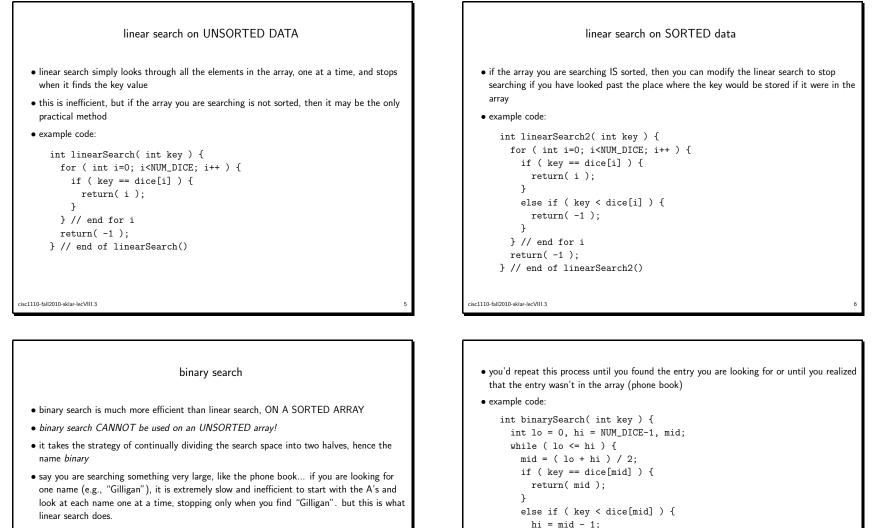
• bubble sort – order $(N-1)^2 = O((N-1)^2)$ because there is one pass made for each element in the array minus 1, and for each pass, the algorithm compares each element in the array to its neighbor, starting with the first element in the array and ending with the second to last element (which takes O(N-1))

searching algorithms

- \bullet often, when you have data stored in an array, you need to locate an element within that array
- this is called searching
- typically, you search for a *key* value (simply the value you are looking for) and return its *index* (the location of the value in the array)
- as with sorting, there are many searching algorithms...
- we'll study the following:
 - sequential or linear search
 - \ast linear search on unsorted data
 - * linear search on sorted data
 - $\ {\rm binary} \ {\rm search}$

cisc1110-fall2010-sklar-lecVIII.3

cisc1110-fall2010-sklar-lecVIII.3



- binary search acts much like you'd act if you were looking up "Gilligan" in the phone book
 - you'd open the book somewhere in the middle, then determine if "Gilligan" appears before or after the page you have opened to
 - if "Gilligan" appears after the page you've selected, then you'd open the book to a later page
 - If "Gilligan" appears before the page you've selected, then you'd open the book to an earlier page

```
cisc1110-fall2010-sklar-lecVIII.3
```

```
} // end of binarySearch()
cisc1110-fall2010-sklar-lecVIII.3
```

7

}

else {

} // end while

return(-1);

lo = mid + 1: