Lecture 6
Searching

Programming Applications (CC213)

AASTMT

Searching

- Searching is the process of determining if a *target* item is present in a list of items, and locating it
- A typical searching algorithm over an array returns the array index where the target item is found, or –1 if it is not found
- We will examine two specific algorithms:
  - Linear search
  - Binary search

Linear Search

- Linear search is also called sequential search
- The approach of linear search:
  - look at each item in turn, beginning with the first one, until either you find the target item or you reach the end of the array

Linear Search

```
int Found = 0;
for ( i=0 ; i < 100 ; i++)
  {
     if ( A[i] == target )
         {
             printf(“ Found it ! “);
             found = 1;
             break ;
         }
   }
if (found ==0)
       printf(“ Looked all over, Not here “);
```

Linear Search

- An example:

3 9 6 1 2 23 8

How many array elements are compared with the target value if the target is 3? Answer: 1

How many array elements are compared with the target value if the target is 2? Answer: 5

How many array elements are compared with the target value if the target is 8? Answer: 7

How many array elements are compared with the target value if the target is 5? Answer: 7

Binary Search

- If an array is ***sorted***
- Check whether the middle element is equal to target. If so, it done.
- Otherwise, determine which half of array the target should be in
- Perform binary search again on subarray by comparing its middle element with target
- Repeatedly divide array in half, until desired item is found, or you have a subarray contains only one element not equal to target, in which case target is not found.

Binary Search

- Example: target value is 25

  Array = **1 3 4 7 7 9 12 17 20 21 25 34 34 39 41**

  Middle element = 17 < 25, 25 should be in second half

  Subarray = 20 21 25 34 34 39 41

  Middle element = 34 > 25, 25 should be in first half

  Subarray = 20 21 25

  Middle element = 21 < 25, 25 should be in second half

  Subarray = 25

  Middle element = 25, done!

Binary Search

Example: target value is 6

Array = 1 3 4 7 7 9 12 17 20 21 25 34 34 39 41

Middle element = 17 > 6, 6 should be in first half

Subarray = 1 3 4 7 7 9 12

Middle element = 7 > 6, 6 should be in first half

Subarray = 1 3 4

Middle element = 3 < 6, 6 should be in second half

Subarray = 4

Middle element = 4 != 6, failed!

Binary Search

```
int low = 0;
int high = 99;
int found = 0;
int mid ;

while ((low <= high) && (found==0)) {
   mid = (low + high) / 2;

  if (target == A[mid])
     {  printf("  Found it ");
        found = 1 ;
     }
  else if (target < A[mid])
         high = mid - 1; /* consider 1st half of array */
  else
    low = mid + 1;  /* consider 2nd half of array */
}

if (found == 0 )
     printf(" Definitely NOT here ");
```

- Example: tracing binary search

2 3 7 7 9 15 16 18 20 21

target = 9, # comparisons = 1			
Iteration	low	mid	high
1	0	4	9

target = 7, # comparisons = 3			
Iteration	low	mid	high
1	0	4	9
2	0	1	3
3	2	2	3

Example: tracing binary search

2 3 7 7 9 15 16 18 20 21

target = 21, # comparisons = 4			
Iteration	low	mid	high
1	0	4	9
2	5	7	9
3	8	8	9
4	9	9	9

target = 22, # comparisons = 4			
Iteration	low	mid	high
1	0	4	9
2	5	7	9
3	8	8	9
4	9	9	9
	10		9

The End