Problem Solving Through Programming in C

Tutorial Session 8

Prof. Anupam Basu
Dept. of Computer Science & Engg.
IIT Kharagpur

Siddhant Mohapatra
PMRF Scholar
IIT Madras

Macros and pre-processor directives

Q. What will be the output of the following C code?

```
#include <stdio.h>
#if A == 1
   #define B 0
#else
   #define B 1
#endif
int main()
{
   printf("%d", B);
   return 0;
}
```

a) 0
b) 1
c) 01
d) None of the above

Q. Which of the following are C pre-processor directive?

a) #if
b) #endif
c) #ifdef
d) #define

#ifdef
#undef
#define

Miscellaneous

Q. What will be the output?

```
#include <stdio.h>
#define a 10
int main()
{
  printf("%d ",a);
  int a=50;
  printf("%d ",a);
  return 0;
}
```

#undef a

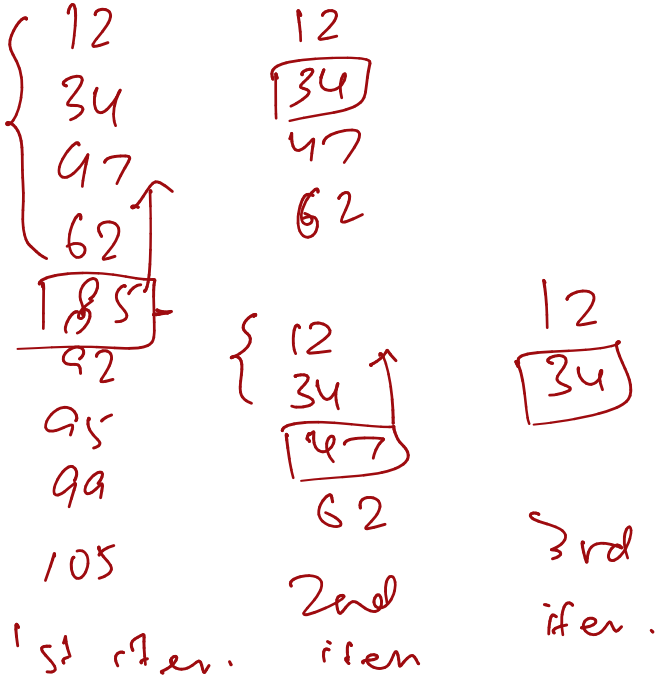
illegal statement.

#redef a 50

a) 10 10
b) 10 50
c) 50 50
d) Compilation error

Q. Given an array arr = {12, 34, 47, 62, 85, 92, 95, 99,105} and key = 34; what are the mid values (corresponding array elements) generated in the first and second iterations of binary search?

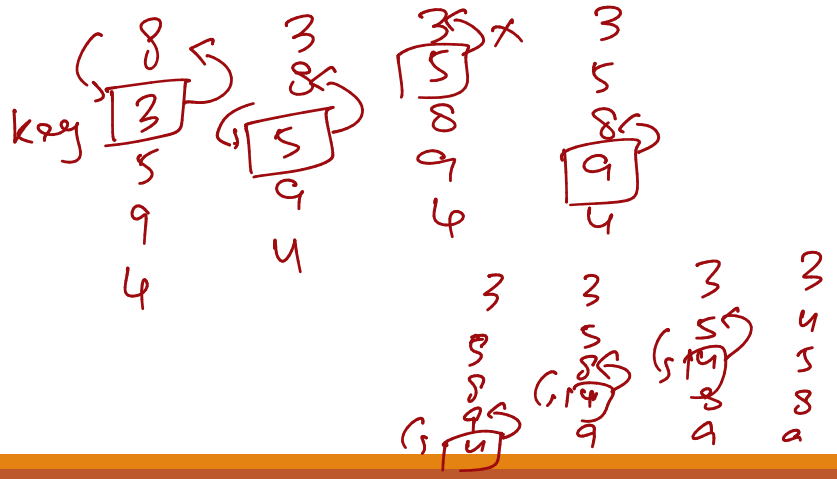
a) 85 and 12
b) 85 and 34
c) 62 and 34
d) 62 and 47



Searching and sorting

Q. When is the Binary Search best applied to an array?

a) For array of very large size
b) When array is sorted
c) When array elements are of mixed data type
d) When array is unsorted



Q. What is the correct order of insertion sort (in ascending order) of the array arr[]={8 3 5 9 4}?

a) {3 8 5 9 4}--> {3 5 8 9 4}--> {3 4 5 8 9}
b) {3 8 5 9 4}--> {3 5 8 9 4}--> {3 5 8 4 9}--> {3 5 4 8 9}-->{3 4 5 8 9}
c) {3 8 5 9 4}-->{3 4 8 5 9}-->{3 4 5 8 9}--> {3 4 5 8 9}-->{3 4 5 8 9}
d) {8 3 5 4 9}-->{8 3 4 5 9}-->{3 4 5 8 9}

Searching and sorting

What is the recurrence relation for the linear search recursive algorithm?

a) T(n-2)+c
b) 2T(n-1)+c
c) T(n-1)+c
d) T(n+1)+c

Consider an array of elements A[7]= {10,4,7,23,67,12,5}, what will be the resultant array A after third pass of insertion sort.

a) 67,12,10,5,4,7,23
b) 4,7,10,23,67,12,5
c) 4,5,7,67,10,12,23
d) 10,7,4,67,23,12,5

Searching and sorting

Q. Which of the following input will give worst case time complexity for selection sort to sort an array in ascending order?

I. 1,2,3, 4, 5, 6, 7, 8
II. 8,7,6, 5, 4, 3, 2, 1,
III. 8,7,5,6,3,2,1,4

a) I
b) II
c) II and III
d) I,II and III

sort in ascending order.

worst case time ⇒ array is descending order
best case time ⇒ array is in ascending order.

→ most no. of operations reqd.
→ least no. of operations reqd.

Q. A sorting technique is called stable if:

a) It takes O(nlog n)time
b) It maintains the relative order of occurrence of non-distinct elements
c) It uses divide and conquer paradigm
d) It takes O(n) space

8 5 2_A 9 6 2_B 3

stable

9 8 6 5 3 2_A 2_B

9 8 6 5 3 2_B 2_A

unstable

Searching and sorting

Q. What is the best case complexity of ordered linear search and worst case complexity of selection sort respectively?

a) $O(1)$, $O(n^2)$
b) $O(\log n)$, $O(1)$,
c) $O(n)$, $O(\log n)$
d) $O(n^2)$, $O(n \log n)$

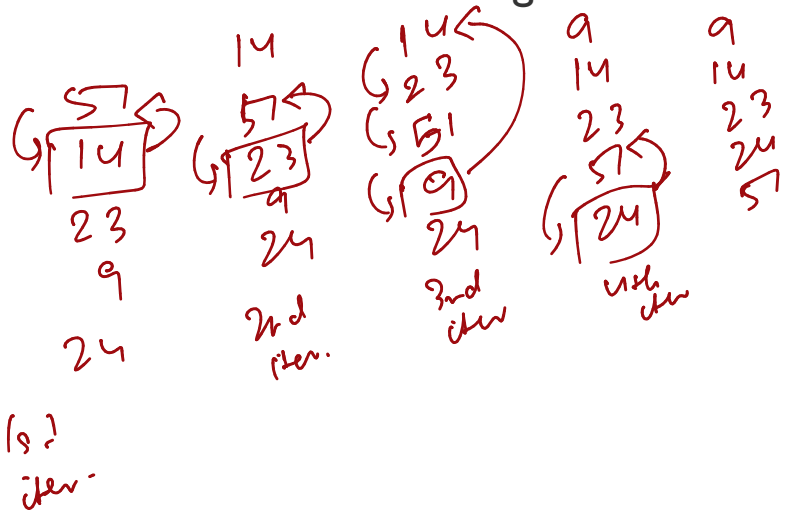
Q. The average case occurs in the Linear Search Algorithm when

a) The item to be searched is in some where middle of the Array
b) The item to be searched is not in the array
c) The item to be searched is in the last of the array
d) The item to be searched is either in the last or not in the array

Searching and sorting

Q. Consider the array arr[]= {51, 14, 23, 9, 24} apply the insertion sort to sort the array . Consider the cost associated with each sort is 1 rupee, what is the total cost of the insertion sort for sorting the entire array?

a) 5
b) 2
c) 3
d) 4



Q. Linear Search can be categorized into which of the following?

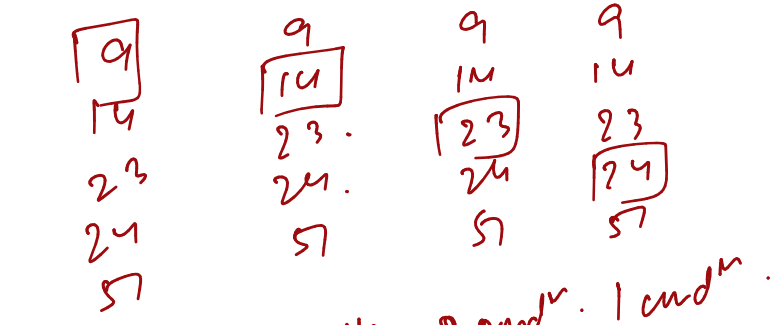
a) Brute Force algorithm → goes through all elements
b) Dynamic programming
c) Greedy algorithm → takes the most optimal value at every step
d) Divide and conquer algorithm

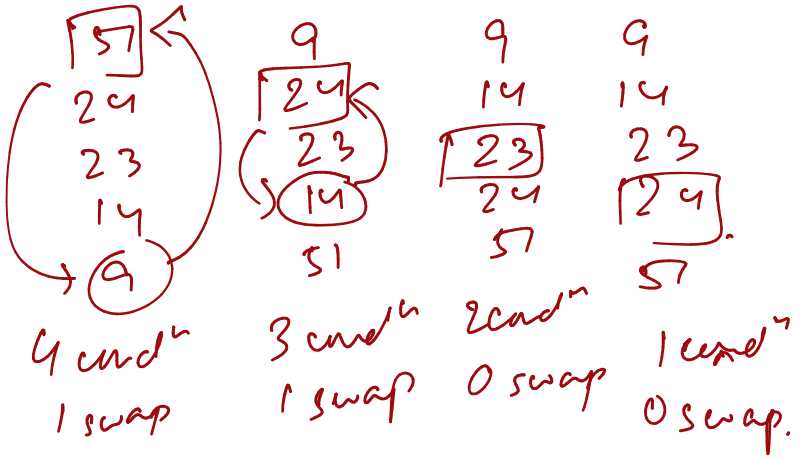
eg- merge sort

eg- selection sort.

Sorting algorithms

Derive the best and worst case complexity of classical selection sort.





N elems
Total no. of ops = $N-1 + N-2 + N-3 + \ldots 1$
$= \frac{N(N-1)}{2} = O(N^2)$

Total no. of ops = $N-1 + N-2 + N-3 + \ldots 1$
$\therefore \frac{N(N-1)}{2} \sim (\leq N) \sim O(N^2)$

Sorting algorithms

Derive the best and worst case complexity of insertion sort.