Week 10 : Assignment 10

Assignment submitted on 2023-09-28, 09:56 IST

1)

Bisection method is used to find

a) Derivative of a function at a given point
b) Numerical integration of a function within a range
c) The root of a function
d) None of the above

a) Option (a)
b) Option (b)
c) Option (c)
d) Option (d)

Yes, the answer is correct.
Score: 1

Accepted Answers:
c) Option (c)

2)

In ....., the search starts at the beginning of the list and checks every element in the list.

a) Linear search
b) Binary search
c) Hash search
d) Binary tree search

a) Option (a)
b) Option (b)
c) Option (c)
d) Option (d)

Yes, the answer is correct.
Score: 1

Accepted Answers:
a) Option (a)

3)

What is the worst-case time complexity of Linear Search?

a) O(1)
b) O(logn)
c) O(n)
d) $O(n^2)$

a) Option (a)
b) Option (b)
c) Option (c)
d) Option (d)

Yes, the answer is correct.
Score: 1

Accepted Answers:
c) Option (c)

4)

What is the worst-case complexity of bubble sort?

a) O(N log N)
b) O(log N)
c) O(N)
d) $O(N^2)$

a) Option (a)
b) Option (b)
c) Option (c)
d) Option (d)

Yes, the answer is correct.
Score: 1

Accepted Answers:
d) Option (d)

5)

What maximum number of comparisons can occur when a bubble sort is implemented? Assume there are n elements in the array.

a) (1/2) (n-1)
b) (1/2) n(n-1)
c) (1/4) n(n-1)
d) None of the above

a) Option (a)
b) Option (b)
c) Option (c)
d) Option (d)

Yes, the answer is correct.
Score: 1

Accepted Answers:
b) Option (b)

6)

What are the correct intermediate steps of the following data set when it is being sorted with the bubble sort? 7,4,1,8,2

a) 4,7,1,8,2→4,1,7,2,8→4,1,2,7,8→1,4,2,7,8→1,2,4,7,8
b) 4,7,1,8,2→4,1,7,8,2→4,1,7,2,8→1,4,7,2,8→1,4,2,7,8→1,2,4,7,8
c) 4,7,1,8,2→1,4,7,8,2→1,4,2,7,8→1,2,4,7,8
d) 4,7,1,8,2→4,7,1,2,8→1,4,7,2,8→1,4,2,7,8→1,2,4,7,8

a) Option(a)
b) Option(b)
c) Option(c)
d) Option(d)

Yes, the answer is correct.
Score: 1

Accepted Answers:
b) Option(b)

7)

What is the main disadvantage of the Bisection Method?

a) It is computationally expensive
b) It cannot find complex roots
c) It requires the function to be differentiable
d) It is not guaranteed to converge

a) Option (a)
b) Option (b)
c) Option (c)
d) Option (d)

No, the answer is incorrect.
Score: 0

Accepted Answers:
b) Option (b)

8)

What will be the output of the following snippet?

```
int arr[] = {10, 20, 30, 40, 50};
int *ptr1 = arr;
int *ptr2 = ptr1 + 3;
printf("%d", *ptr2 - *ptr1);
```

Hint

Yes, the answer is correct.
Score: 1

Accepted Answers:
(Type: Numeric) 30

9)

What is the solution of the equation given below using the Bisection Method up to four decimal places? (Consider the root lying on positive quadrant only and compute the root till five iterations only)

$$f(x) = xe^{2x} - 3x^2 - 5$$

Hint

No, the answer is incorrect.
Score: 0

Accepted Answers:
(Type: Numeric) 1.0312

10)

What will be the output?

```
#include <stdio.h>
int main(void)
{
int a[] = {10, 12, 6, 7, 2};
int i, *p;
p=a+4;
for(i=0; i<5; i++)
printf("%d ", p[-i]);
return 0;
}
```

a) Option (a)
b) Option (b)
c) Option (c)
d) Option (d)

No, the answer is incorrect.
Score: 0

Accepted Answers:
d) Option (d)

Week 10 : Programming Assignment 01

Write a C program to find the root of the equation using bisection method for different values of allowable error of the root.

$$f(x) = 2x^3 - 3x - 5$$

Private Test cases used for evaluation	Input	Expected Output	Actual Output	Status
Test Case 1	0.01	Root = 1.7266	Root = 1.5000\n	Wrong Answer

Assignment submitted on 2023-10-05, 23:30 IST

Your last recorded submission was :

```c
#include<stdio.h>
float fun (float x); //Function fun returns the function value of f(x)
void bisection (float *x, float a, float b, int *itr); // This function computes the root of f(x) using bisection method

int main ()
{
    int itr = 0, maxmitr=10;
    float x, a=1.0, b=2.0, allerr, x1; // x is the value of root in each iteration, x1 is the final value of the root
   // a and b are the initial range for calculating the root using bisection method

scanf("%f", &allerr);  // allerr is the allowable error taken from test case data
    bisection (&x, a, b, &itr);

/* Use the printf statement as given below to print the root
printf("Root = %1.4f\n", x1); */
do{
   if(fun(a)*fun(b) < 0)
     b=x;
   else
     a=x;
   bisection(&x1,a,b,&itr);
   if(x1-x < allerr)
   {
     printf("Root = %1.4f\n", x);
     return 0;
   }
   x=x1;
}
while(itr<maxmitr);
}


float fun(float x)
{
  return (2*x*x*x - 3*x - 5);
}

void bisection (float *x, float a, float b, int *itr){
    *x = (a+b)/2;
    ++(*itr);
}
```

Week 10 : Programming Assignment 02

Write a C code to check if a 3 x 3 matrix is invertible. A matrix is not invertible if its determinant is 0.

Private Test cases used for evaluation	Input	Expected Output	Actual Output	Status
Test Case 1	4 5 6 7 8 9 1 2 3	The given matrix is not invertible	The given matrix is not invertible	Passed
Test Case 2	1 2 3 0 1 4 5 6 0	The given matrix is invertible	The given matrix is invertible	Passed

Assignment submitted on 2023-10-04, 12:18 IST

Your last recorded submission was :

```c
#include<stdio.h>
 int main()
{
  int a[3][3], i, j;
  long determinant;
 //  9 elements of matrix is taken as input from test data
  for(i = 0 ; i < 3; i++)
      for(j = 0; j < 3; j++)
           scanf("%d", &a[i][j]);

/*Use the printf statements as:
printf("The given matrix is not invertible");
printf("The given matrix is invertible");
*/
determinant = a[0][0] * ((a[1][1]*a[2][2]) - (a[2][1]*a[1][2])) -a[0][1] * (a[1][0]
   * a[2][2] - a[2][0] * a[1][2]) + a[0][2] * (a[1][0] * a[2][1] - a[2][0] * a[1][1]);

if(determinant == 0)
  printf("The given matrix is not invertible");

else
  printf("The given matrix is invertible");

}
```

Week 10 : Programming Assignment 03

Write a C program to sort a given 1D array using pointer in ascending order.

Private Test cases used for evaluation	Input	Expected Output	Actual Output	Status
Test Case 1	8 90 70 30 -10 -40 20 100 50	-40\n -10\n 20\n 30\n 50\n 70\n 90\n 100	-40\n -10\n 20\n 30\n 50\n 70\n 90\n 100\n	Passed

Assignment submitted on 2023-10-04, 12:25 IST

Your last recorded submission was :

```c
#include <stdio.h>
int main()
{
    int a[100],i, n;
    scanf("%d",&n);  //Number of elements of the array is taken from the test case data

   for (i=0; i<n; i++)
    {
        scanf("%d",a+i); // Input the array elements
    }
int j,t;
  for (i = 0; i < n; i++) {

        for (j = i + 1; j < n; j++) {

            if (*(a + j) < *(a + i)) {

                t = *(a + i);
                *(a + i) = *(a + j);
                *(a + j) = t;
            }
        }
    }
//   Printing sorted array in ascending order
    for (i=0; i<n; i++)
    {
        printf("%d\n",*(a+i));
    }
    return 0;
   }
```

Week 10 : Programming Assignment 04

Write a C program to sort a 1D array using pointer by applying Bubble sort technique.

Private Test cases used for evaluation	Input	Expected Output	Actual Output	Status
Test Case 1	7 70 40 80 10 200 30 60	10\n 30\n 40\n 60\n 70\n 80\n 200	10\n 30\n 40\n 60\n 70\n 80\n 200\n	Passed

Assignment submitted on 2023-10-04, 12:28 IST

Your last recorded submission was :

```c
#include<stdio.h>
void sort(int *a, int n);
int main()
{
    int a[20];
    int n,i;
    scanf("%d",&n); // Enter number of elements to sort is taken from test case data

    for(i=0;i<n;i++)
    {
        scanf("%d",&a[i]); // The elements of the array is taken from the test data
    }

sort(a, n); // Calling the sorting function

    //Printing the sorted array
    for(i=0;i<n;i++)
    {
        printf("%d\n",a[i]);
    }
   return 0;
}
void sort(int *a, int n)
{
    int i, j, t;

    // Sort the numbers using pointers
    for (i = 0; i < n; i++) {

        for (j = i + 1; j < n; j++) {

            if (*(a + j) < *(a + i)) {

                t = *(a + i);
                *(a + i) = *(a + j);
                *(a + j) = t;
            }
        }
    }
}
```