214189E – SENARATHNA G.G.P.C.

B21-S2-IN1111 - Data Structures and Algorithms.

Sorting Algorithms,

* Insertion Sort
* #include <stdio.h>  
    
  int main() {  
   int arr[] = {6,5,3,1,8,7,2,4};  
   int temp,size;  
    
   size = sizeof(arr)/sizeof(arr[0]);  
    
   for(int i = 1;i < size;i++){  
   temp = arr[i];  
   // printf("%d \n", temp);  
   for(int j = i - 1;j >= 0;j--){  
   if(temp < arr[j]){  
   arr[j + 1] = arr[j];  
   // printf("%d ", arr[j + 1]);  
   if(j == 0){  
   arr[j] = temp;  
   // printf("%d \n", arr[j]);  
   }  
    
   }  
   else{  
   arr[j + 1] = temp;  
   break;  
   }  
   }  
   }  
   for(int i = 0;i < size;i++){  
   printf("%d ", arr[i]);  
   }  
    
   return 0;  
  }

execution time,

* Best – 109 ms
* Worst – 136 ms
* Average – 125 ms
* Selection Sort,
* #include <stdio.h>  
    
  int main(){  
   int arr[] = {8,5,2,6,9,3,1,4,0,7};  
   int min, k, size;  
    
   size = sizeof(arr)/sizeof(arr[0]);  
    
   for(int i = 0;i < size;i++){  
   min = arr[i];  
   k = 0;  
   for(int j = i + 1;j < size;j++){  
   if(min > arr[j]){  
   min = arr[j];  
   k = j;  
   }  
   }  
   if (k > 0){  
   arr[k] = arr[i];  
   arr[i] = min;  
   }  
    
   }  
   for(int i = 0;i < size;i++){  
   printf("%d ", arr[i]);  
   }  
    
   return 0;  
  }

execution time,

* Best – 113 ms
* Worst – 110 ms
* Average – 110 ms
* Bubble Sort,
* #include <stdio.h>  
    
  int main(){  
   int arr[] = {6,5,3,1,8,7,2,4};  
   int temp, size;  
    
   size = sizeof(arr)/sizeof(arr[0]);  
    
   for(int i = 0;i < size;i++){  
   temp = 0;  
   for(int j = 0;j < size - i;j++){  
   if(arr[j] > arr[j + 1]){  
   temp = arr[j];  
   arr[j] = arr[j + 1];  
   arr[j + 1] = temp;  
   }  
   }  
   }  
   for(int i = 0;i < size;i++){  
   printf("%d ", arr[i]);  
   }  
    
   return 0;  
  }

execution time,

* Best – 93 ms
* Worst – 94 ms
* Average – 94 ms

Randomly Generate

* Bubble Sort
* #include <stdio.h>  
  #include <stdlib.h>  
    
  int main(){  
   int arr[1000];  
   int temp;  
    
   for(int i = 0;i < 1000;i++){  
   arr[i] = rand();  
   }  
    
   for(int i = 0;i < 1000;i++){  
   temp = 0;  
   for(int j = 0;j < 1000 - i;j++){  
   if(arr[j] > arr[j + 1]){  
   temp = arr[j];  
   arr[j] = arr[j + 1];  
   arr[j + 1] = temp;  
   }  
   }  
   }  
   for(int i = 0;i < 1000;i++){  
   printf("%d ", arr[i]);  
   }  
    
   return 0;  
  }