|  |  |
| --- | --- |
| Algorithm  [2022] |  |
|  |  |
| [01.09.2022]  Techno India College of Technology  Authored by: Abu Taha Md Farhad  Roll: 31401221052  Reg No: 213141001210016 OF 2021-22  Stream: BCA 2nd year  Subject: Data Structure & Algorithm [BCAC303] |  |

Sorting

# Insertion Sort:

1. Start
2. Input the value of n.
3. Take an array a[] of n elements.
4. Input n elements in array a[].
5. Initialize, i = 2.
6. Go through steps 7, 8, 9, 13 while i <= n.
7. Set j = i – 1.
8. Set key = a[i].
9. Go through steps 10, 11, 12 while key <= a[j] and j >= 1.
10. Set a[j+1] = a[j].
11. Set a[j] = key.
12. Set j = j – 1.

[End of Step 9 while]

1. Set i = i + 1.

[End of Step 6 while]

1. Print n elements of array a[].
2. End.

# Bubble sort:

1. Start
2. Input the value of n.
3. Take an array a[] of n elements.
4. Input n elements in array a[].
5. Initialize, i = 1.
6. Go through steps while i <= n.
7. Initialize, j = i +1.
8. Go through steps while j <= n.
9. If a[i] > a[j]

Then

Initialize, temp = a[i].

Set a[i] = a[j].

Set a[j] = temp.

[End of if]

1. Set j = j +1.

[End of Step 8 while]

1. Set i = i + 1.

[End of Step 6 while]

1. Print n elements of array a[].
2. End.

# Selection Sort:

1. Start.
2. Input the value of n.
3. Take an array a[] size of n elements.
4. Input n elements in array a[].
5. Initialize, i = n -1.
6. Go through steps 7, 8, 9, 10, 13, 14, 15 while i >= 1.
7. Set max = a[0].
8. Set index = 0.
9. Initialize, j = 1.
10. Go through steps 11, 12 while j <= i.
11. If a[j] > max

Then

Set max = a[j].

Set index = j.

[End of if]

1. Set j = j + 1.

[End of Step 10 while]

1. Set a[index] = a[i].
2. Set a[i] = max.
3. Set i = i – 1.

[End of Step 6 while]

1. Print n elements of array a[].
2. End.

# Merge Sort:

1. Start.
2. Input the value of n.
3. Take an array a[] size of n elements.
4. Input n elements in array a[].
5. Initialize beg = 1
6. Initialize end = n
7. MERGE\_SORT(a, beg, end)
8. If beg < end

Then

Set mid = (beg + end)/2

MERGE\_SORT(a, beg, mid)

MERGE\_SORT(a, mid + 1, end)

MERGE (a, beg, mid, end)

[End of if]

1. [End of MERGE\_SORT function]
2. Print n elements of array a[].
3. End.

# Quick Sort:

1. Start.
2. Input the value of n.
3. Take an array a[] size of n elements.
4. Input n elements in array a[].
5. Initialize start = 1
6. Initialize end = n
7. QUICKSORT (a, start, end)
8. If (start < end)

Then

Set p = partition(a, start, end)

QUICKSORT (a, start, p - 1)

QUICKSORT (a, p + 1, end)

[End of if]

1. [End of QUICKSORT function]
2. Print n elements of array a[].
3. End.

Searching

# Linear search:

1. Start
2. Input the value of n.
3. Take an array a[] of n elements.
4. Input n elements in array a[].
5. Take the search value as search.
6. Initialize, i = 1.
7. Go through steps 8, 9 while i <= n.
8. If a[i] == search

Then

Print(search + “ found at position ” + i).

Go to step 11.

[End of if]

1. Set i = i + 1.

[End of while]

1. Print(search + “is not found”).
2. End.

# Binary search:

1. Start
2. Input the value of n.
3. Take an array a[] of n elements.
4. Input n elements in array a[].
5. Take the search value as search.
6. Initialize, beg = 1 and end = n.
7. Go through steps 8, 9 while beg < end
8. Set mid = floor((beg + end) / 2).
9. If a[mid] == search

Then

Print(search + “ found at position ” + i).

Go to step 11.

Else

If a[mid] > search

Then

Set end = mid – 1

Else

Set beg = mid + 1

[End of inner if]

[End of outer if]

[End of while]

1. Print(search + “is not found”).
2. End.

Insertion

# Insert at beginning:

1. Start
2. Input the value of n.
3. Take an array a[] of n elements.
4. Input n elements in array a[].
5. Input the value to be inserted in val.
6. Initialize, i = n.
7. Go through steps 8, 9 while i >= 1.
8. Set a[i+1] = a[i]
9. Set i = i - 1.

[End of while]

1. Set a[1] = val.
2. Set n = n + 1
3. Print n elements of array a[].
4. End.

# Insert at last:

1. Start.
2. Input the value of n.
3. Take an array a[] of n elements.
4. Input n elements in array a[].
5. Input the value to be inserted in val.
6. Set a[n+1] = val.
7. Set n = n + 1
8. Print n elements of array a[].
9. End.

# Insert at any position:

1. Start.
2. Input the value of n.
3. Take an array a[] of n elements.
4. Input n elements in array a[].
5. Input the value to be inserted in val.
6. Input the position where the value to be inserted in pos.
7. Initialize, i = n.
8. Go through steps 9, 10 while i >= pos.
9. Set a[i+1] = a[i].
10. Set i = i – 1.

[End of while]

1. Set a[pos] = val.
2. Set n = n + 1
3. Print n elements of array a[].
4. End.

Deletion

# Delete from beginning:

1. Start.
2. Input the value of n.
3. Take an array a[] of n elements.
4. Input n elements in array a[].
5. Initialize, i = 1
6. Go through steps 7, 8 while i < n.
7. Set a[i] = a[i+1].
8. Set i = i + 1.

[End of while]

1. Set n = n -1
2. Print n elements of array a[].
3. End.

# Delete from last:

1. Start.
2. Input the value of n.
3. Take an array a[] of n elements.
4. Input n elements in array a[].
5. Set n = n -1
6. Print n elements of array a[].
7. End.

# Delete from any position:

1. Start.
2. Input the value of n.
3. Take an array a[] of n elements.
4. Input n elements in array a[].
5. Input the position where from the element to be deleted in pos.
6. Initialize, i = pos.
7. Go through steps 8, 9 while i < n.
8. Set a[i] = a[i+1].
9. Set i = i + 1.

[End of while]

1. Set n = n -1
2. Print n elements of array a[].
3. End.