

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:

- Step 1. Start
- Step 2. Input the value of n .
- Step 3. Take an array $a[]$ of n elements.
- Step 4. Input n elements in array $a[]$.
- Step 5. Initialize, $i = 2$.
- Step 6. Go through steps 7, 8, 9, 13 while $i \leq n$.
- Step 7. Set $j = i - 1$.
- Step 8. Set $\text{key} = a[i]$.
- Step 9. Go through steps 10, 11, 12 while $\text{key} \leq a[j]$ and $j \geq 1$.
- Step 10. Set $a[j+1] = a[j]$.
- Step 11. Set $a[j] = \text{key}$.
- Step 12. Set $j = j - 1$.
- [End of Step 9 while]
- Step 13. Set $i = i + 1$.
- [End of Step 6 while]
- Step 14. Print n elements of array $a[]$.
- Step 15. End.

Bubble sort:

- Step 1. Start
- Step 2. Input the value of n .
- Step 3. Take an array $a[]$ of n elements.
- Step 4. Input n elements in array $a[]$.
- Step 5. Initialize, $i = 1$.
- Step 6. Go through steps while $i \leq n$.
- Step 7. Initialize, $j = i + 1$.
- Step 8. Go through steps while $j \leq n$.
- Step 9. If $a[i] > a[j]$
 - Then
 - Initialize, $temp = a[i]$.
 - Set $a[i] = a[j]$.
 - Set $a[j] = temp$.
 - [End of if]
- Step 10. Set $j = j + 1$.
 - [End of Step 8 while]
- Step 11. Set $i = i + 1$.
 - [End of Step 6 while]
- Step 12. Print n elements of array $a[]$.
- Step 13. End.

Selection Sort:

- Step 1. Start.
- Step 2. Input the value of n .
- Step 3. Take an array $a[]$ size of n elements.
- Step 4. Input n elements in array $a[]$.
- Step 5. Initialize, $i = n - 1$.
- Step 6. Go through steps 7, 8, 9, 10, 13, 14, 15 while $i \geq 1$.
- Step 7. Set $\text{max} = a[0]$.
- Step 8. Set $\text{index} = 0$.
- Step 9. Initialize, $j = 1$.
- Step 10. Go through steps 11, 12 while $j \leq i$.
- Step 11. If $a[j] > \text{max}$
 - Then
 - Set $\text{max} = a[j]$.
 - Set $\text{index} = j$.
 - [End of if]
- Step 12. Set $j = j + 1$.
 - [End of Step 10 while]
- Step 13. Set $a[\text{index}] = a[i]$.
- Step 14. Set $a[i] = \text{max}$.
- Step 15. Set $i = i - 1$.
 - [End of Step 6 while]
- Step 16. Print n elements of array $a[]$.
- Step 17. End.

Merge Sort:

- Step 1. Start.
- Step 2. Input the value of n.
- Step 3. Take an array a[] size of n elements.
- Step 4. Input n elements in array a[].
- Step 5. Initialize beg = 1
- Step 6. Initialize end = n
- Step 7. MERGE_SORT(a, beg, end)
- Step 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]
- Step 9. [End of MERGE_SORT function]
- Step 10. Print n elements of array a[].
- Step 11. End.

Quick Sort:

- Step 1. Start.
- Step 2. Input the value of n.
- Step 3. Take an array a[] size of n elements.
- Step 4. Input n elements in array a[].
- Step 5. Initialize start = 1
- Step 6. Initialize end = n
- Step 7. QUICKSORT (a, start, end)
- Step 8. If (start < end)
 - Then
 - Set p = partition(a, start, end)
 - QUICKSORT (a, start, p - 1)
 - QUICKSORT (a, p + 1, end)
 - [End of if]
- Step 9. [End of QUICKSORT function]
- Step 10. Print n elements of array a[].
- Step 11. End.

Searching

Linear search:

- Step 1. Start
- Step 2. Input the value of n.
- Step 3. Take an array a[] of n elements.
- Step 4. Input n elements in array a[].
- Step 5. Take the search value as search.
- Step 6. Initialize, $i = 1$.
- Step 7. Go through steps 8, 9 while $i \leq n$.
- Step 8. If $a[i] == \text{search}$
 - Then
 - Print(search + " found at position " + i).
 - Go to step 11.
- [End of if]
- Step 9. Set $i = i + 1$.
- [End of while]
- Step 10. Print(search + "is not found").
- Step 11. End.

Binary search:

- Step 1. Start
- Step 2. Input the value of n.
- Step 3. Take an array a[] of n elements.
- Step 4. Input n elements in array a[].
- Step 5. Take the search value as search.
- Step 6. Initialize, beg = 1 and end = n.
- Step 7. Go through steps 8, 9 while beg < end
- Step 8. Set mid = floor((beg + end) / 2).
- Step 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]
 - Step 10. Print(search + “is not found”).
 - Step 11. End.

Insertion

Insert at beginning:

- Step 1. Start
- Step 2. Input the value of n.
- Step 3. Take an array a[] of n elements.
- Step 4. Input n elements in array a[].
- Step 5. Input the value to be inserted in val.
- Step 6. Initialize, $i = n$.
- Step 7. Go through steps 8, 9 while $i \geq 1$.
- Step 8. Set $a[i+1] = a[i]$
- Step 9. Set $i = i - 1$.
- [End of while]
- Step 10. Set $a[1] = val$.
- Step 11. Set $n = n + 1$
- Step 12. Print n elements of array a[].
- Step 13. End.

Insert at last:

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

Insert at any position:

- Step 1. Start.
- Step 2. Input the value of n .
- Step 3. Take an array $a[]$ of n elements.
- Step 4. Input n elements in array $a[]$.
- Step 5. Input the value to be inserted in val .
- Step 6. Input the position where the value to be inserted in pos .
- Step 7. Initialize, $i = n$.
- Step 8. Go through steps 9, 10 while $i \geq pos$.
- Step 9. Set $a[i+1] = a[i]$.
- Step 10. Set $i = i - 1$.
[End of while]
- Step 11. Set $a[pos] = val$.
- Step 12. Set $n = n + 1$
- Step 13. Print n elements of array $a[]$.
- Step 14. End.

Deletion

Delete from beginning:

- Step 1. Start.
- Step 2. Input the value of n .
- Step 3. Take an array $a[]$ of n elements.
- Step 4. Input n elements in array $a[]$.
- Step 5. Initialize, $i = 1$
- Step 6. Go through steps 7, 8 while $i < n$.
- Step 7. Set $a[i] = a[i+1]$.
- Step 8. Set $i = i + 1$.
[End of while]
- Step 9. Set $n = n - 1$
- Step 10. Print n elements of array $a[]$.
- Step 11. End.

Delete from last:

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

Delete from any position:

- Step 1. Start.
- Step 2. Input the value of n.
- Step 3. Take an array a[] of n elements.
- Step 4. Input n elements in array a[].
- Step 5. Input the position where from the element to be deleted in pos.
- Step 6. Initialize, i = pos.
- Step 7. Go through steps 8, 9 while i < n.
- Step 8. Set a[i] = a[i+1].
- Step 9. Set i = i + 1.
[End of while]
- Step 10. Set n = n - 1
- Step 11. Print n elements of array a[].
- Step 12. End.