Merge Sort with Time complexity analysis

Design and Analysis of Algorithms

Session Learning Outcome-SLO

- At the end of this session, you should be able to perform merge sort and also evaluate its time complexity

Introduction

- Attributed to a Hungarian mathematician John van Neumann
- Used for sorting unordered arrays
- Uses divide-and-conquer strategy
- **1st phase** is to **divide** the array of numbers into 2 equal parts
  - If necessary, these subarrays are divided further
- **2nd phase** is the **conquer** part
  - Involves sorting of subarrays recursively and combine the sorted arrays to give the final sorted list

Merge sort – informal procedure

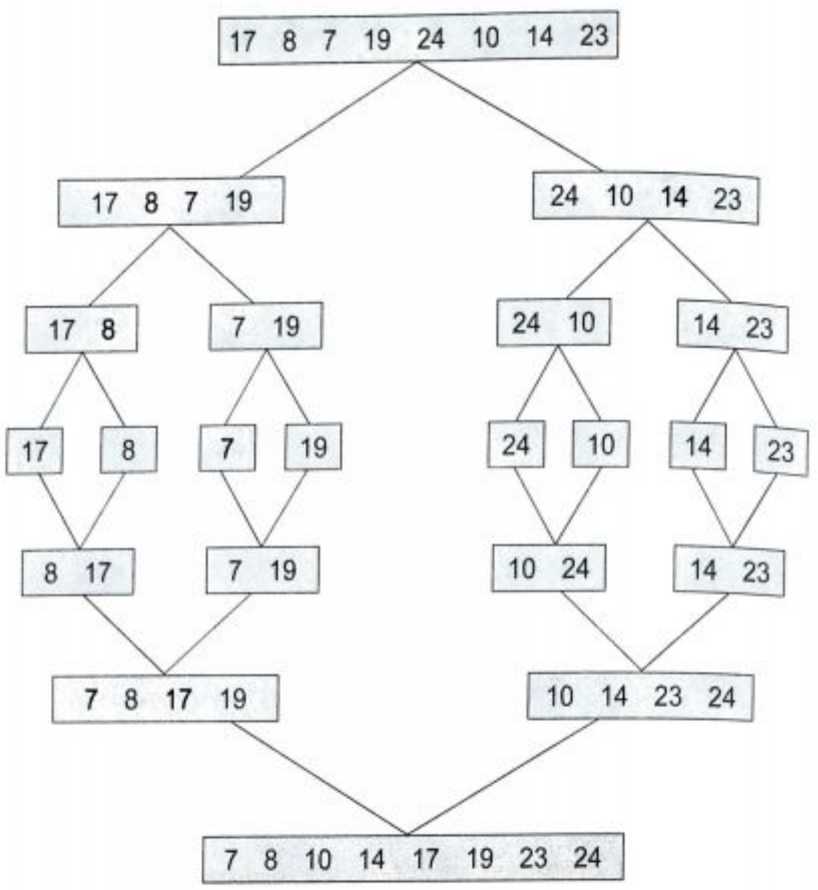
Step1: Divide an array into subarrays B and C

Step 2: Sort subarray D recursively; this yields B sorted subarray

Step 3: Sort subarray C recursively; this yields C sorted subarray

Step 4: Combine B and C sorted subarrays to obtain the final sorted array A

MergeSort- An Example



Algorithm mergesort(A[first .. Last])

Input: Unsorted array A with first =1 and last=n

Output: Sorted array A

```
Begin
    if (first == last) then
        return A[first]
    else
        mid = (first+last)/2
    for i ∈ {1,2, …, mid}
        B[i] = A[i]
    End for
    for i ∈ {mid +1, …, n}
        C[i] = A[i]
    End for
    mergesort(B[1 .. mid])
    mergesort(C[mid + 1 .. n])
    merge(B,C,A)
End
```

Algorithm merge(B,C,A)

Input: Two sorted arrays B and C

Output: Sorted array A

```
Begin
i=1
j=1
k=1
m = length(B)
n = length(C)
while ((i<=m) and (j <= n)) do
if(B[i] < C[j]) then
A[k] = B[i]
i=i+1
else
A[k] = C[i]
j=j+1
End if
k=k+1
end while
if (i > m) then
while k <= m + n do
A[k] = C[j]
j,k = j+1, k+1
end while
else if (j < n) then
while (k <= m+n) do
A[k] = B[j]
i,k = i+1,k+1
end while
end if
return (A)
end
```

Complexity Analysis

- T(n) = $2T\left(\frac{n}{2}\right) + n - 1, for\ n \geq 2$

  = 1, for n=2

  = 0, when n<2

Summary

- Uses divide and conquer strategy
- a comparison based sort
- out of place merge sort
- Stable Algorithm
- Merging method is used
- Variants of merge sort

    in place

    bottom up merge sort

    top down merge sort

Questions

- Why is merge sort an out-of-place sorting technique?
- Does it use recursive procedure?
- What are the best, worst and average case time complexities?

Reference

- S. Sridhar, Design and Analysis of Algorithms, Oxford University Press, 2015

Thank You