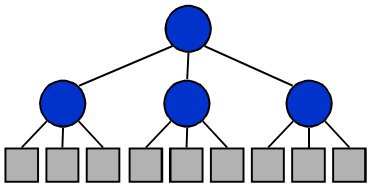
CSE 203: Data Structures and Algorithms-I

Divide-and-Conquer Technique
Arrays: Merge Sort, Quick Sort

Dr. Md. Abul Kashem Mia, Professor, CSE Dept and Pro-Vice Chancellor, UIU

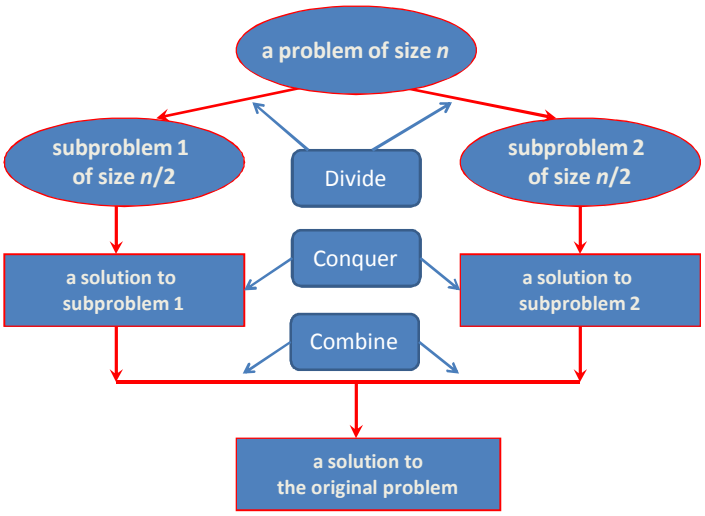
Divide-and-Conquer Technique

- Divide-and-Conquer is a general algorithm design paradigm:
  - Divide the problem into a number of subproblems that are smaller instances of the same problem
  - Conquer the subproblems by solving them recursively
  - Combine the solutions to the subproblems into the solution for the original problem
- The base case for the recursion are subproblems of constant size
- Analysis can be done using **recurrence equations**



Dr. Md. Abul Kashem Mia, Professor, CSE Dept and Pro-Vice Chancellor, UIU

Divide-and-Conquer



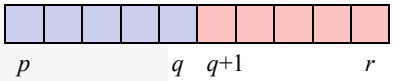
Merge Sort and Quick Sort

Two well-known sorting algorithms adopt this divide-and-conquer strategy

- Merge sort
  - Divide step is trivial – just split the list into two equal parts
  - Work is carried out in the conquer step by merging two sorted lists
- Quick sort
  - Work is carried out in the divide step using a pivot element
  - Conquer step is trivial

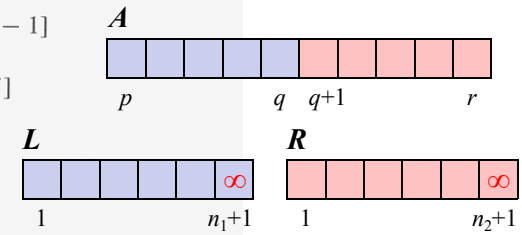
Merge Sort: Algorithm

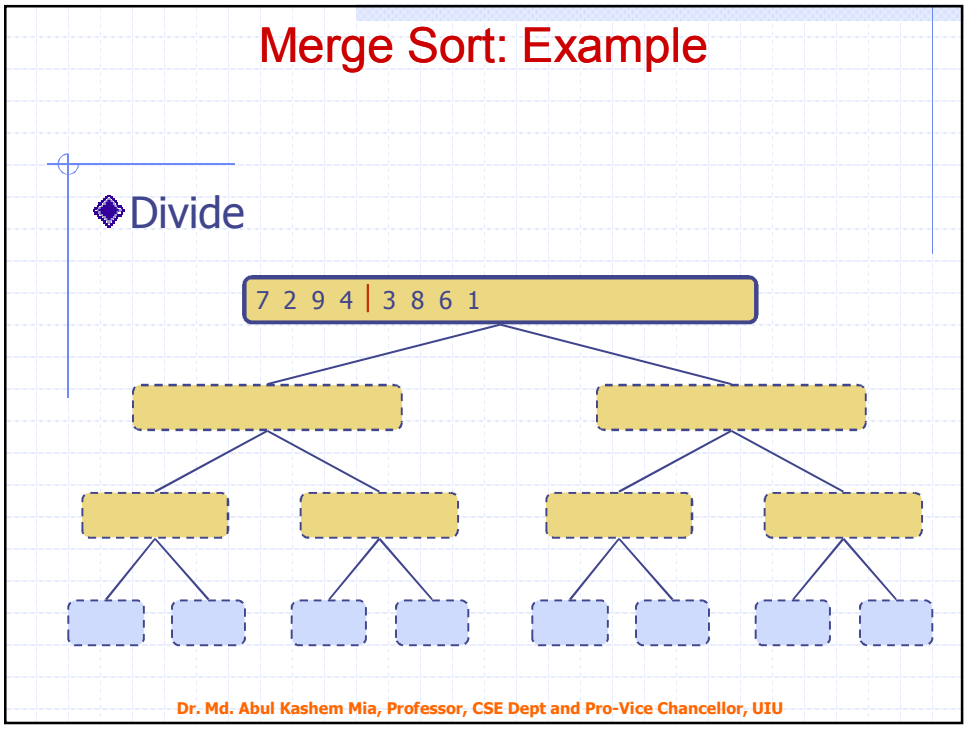
```
MERGE-SORT(A, p, r)
1  if p < r
2     then q ← ⌊(p + r)/2⌋
3          MERGE-SORT(A, p, q)
4          MERGE-SORT(A, q + 1, r)
5          MERGE(A, p, q, r)
```

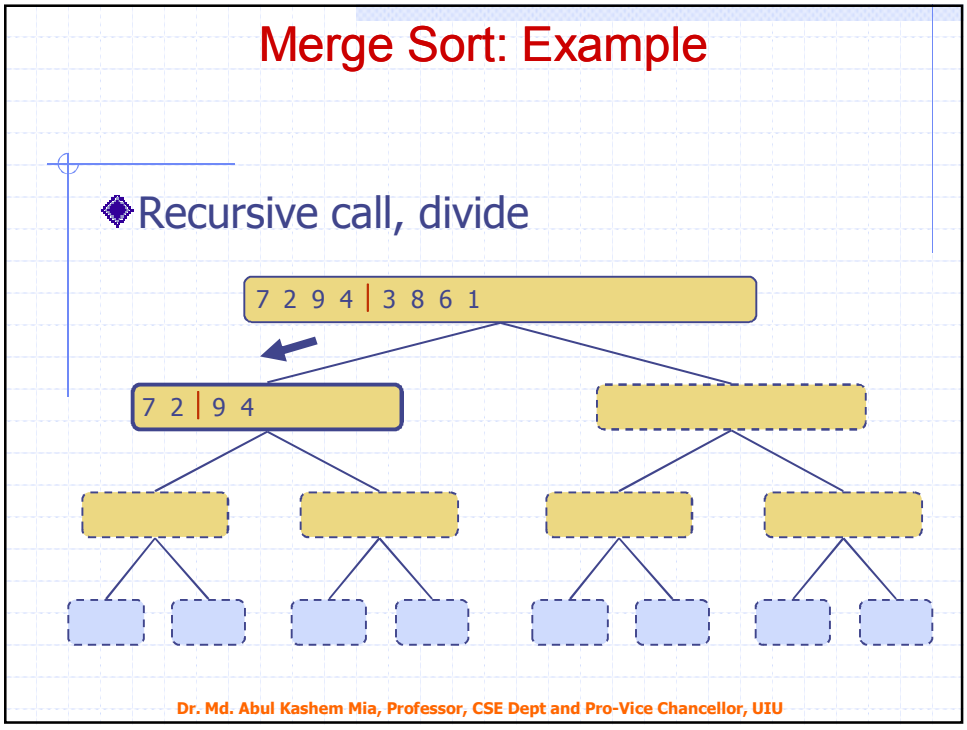


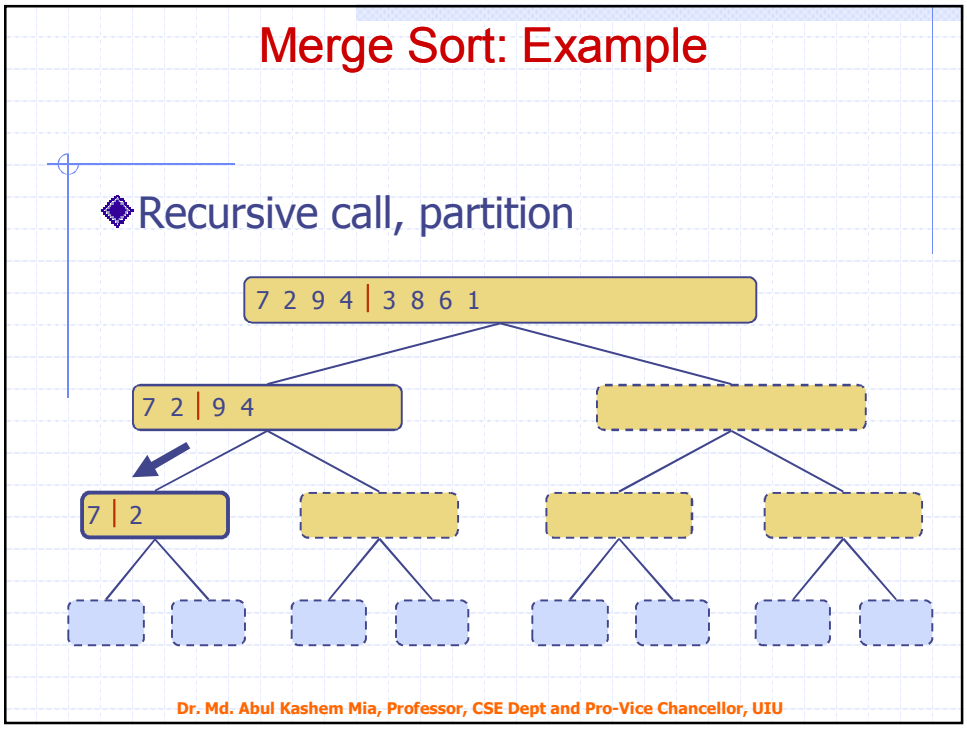
Merge Sort: Algorithm

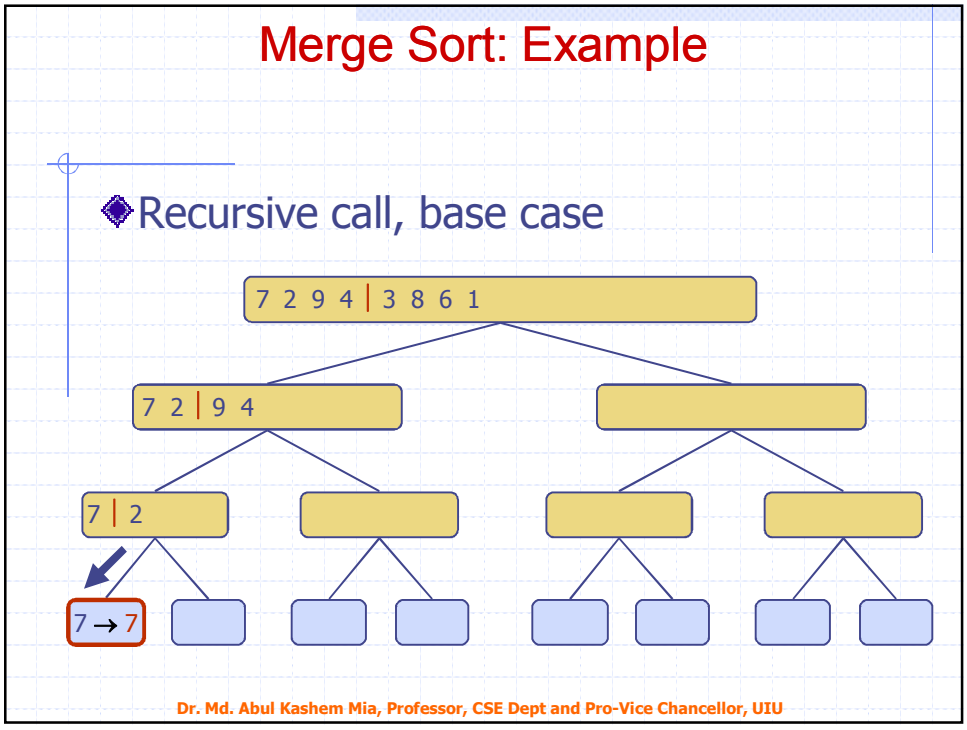
```
MERGE(A, p, q, r)
1  n1 ← q − p + 1
2  n2 ← r − q
3  create arrays L[1 .. n1 + 1] and R[1 .. n2 + 1]
4  for i ← 1 to n1
5       do L[i] ← A[p + i − 1]
6  for j ← 1 to n2
7       do R[j] ← A[q + j]
8  L[n1 + 1] ← ∞
9  R[n2 + 1] ← ∞
10 i ← 1
11 j ← 1
12 for k ← p to r
13      do if L[i] ≤ R[j]
14           then A[k] ← L[i]
15                i ← i + 1
16           else A[k] ← R[j]
17                j ← j + 1
```

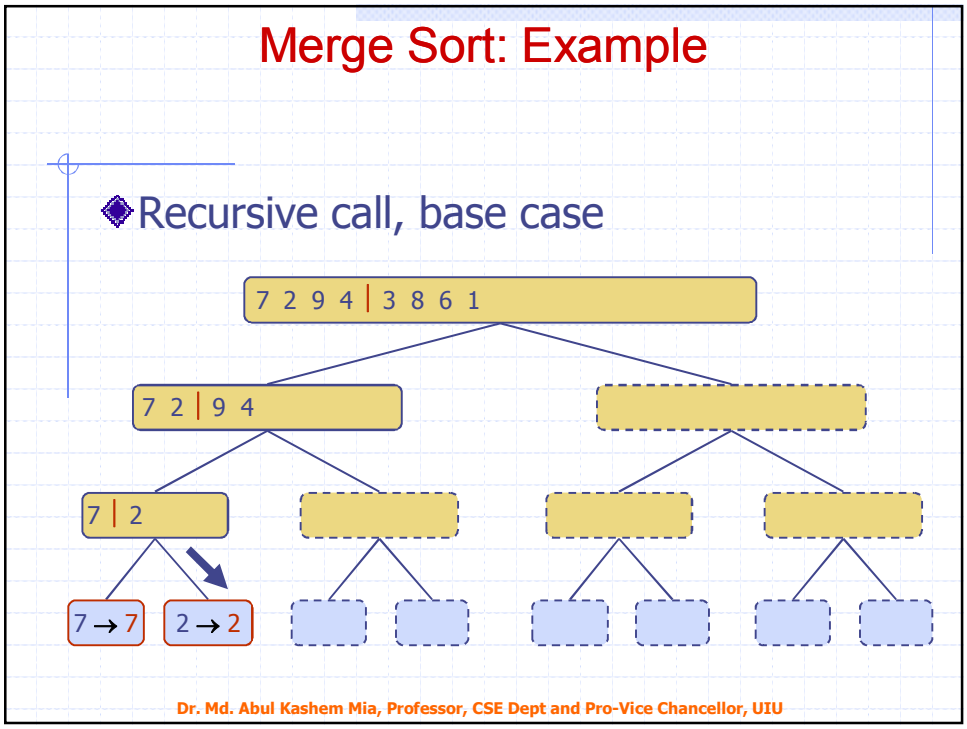


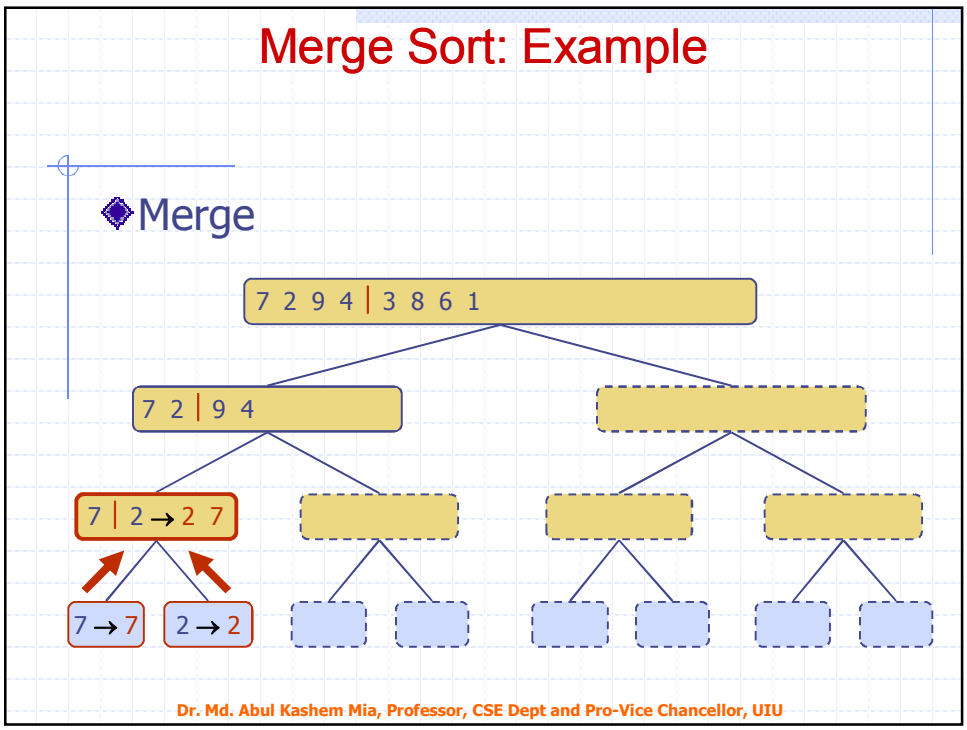

Merge Sort: Example
Divide
7 2 9 4 | 3 8 6 1
Dr. Md. Abul Kashem Mia, Professor, CSE Dept and Pro-Vice Chancellor, UIU

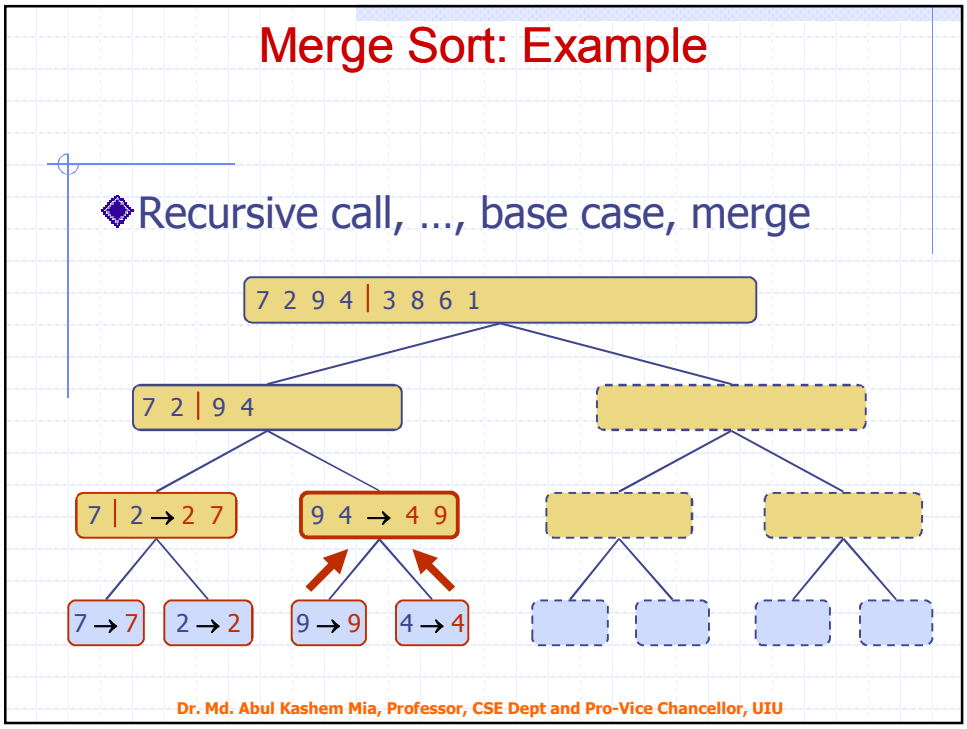

Merge Sort: Example
Recursive call, divide
7 2 9 4 | 3 8 6 1
7 2 | 9 4
Dr. Md. Abul Kashem Mia, Professor, CSE Dept and Pro-Vice Chancellor, UIU

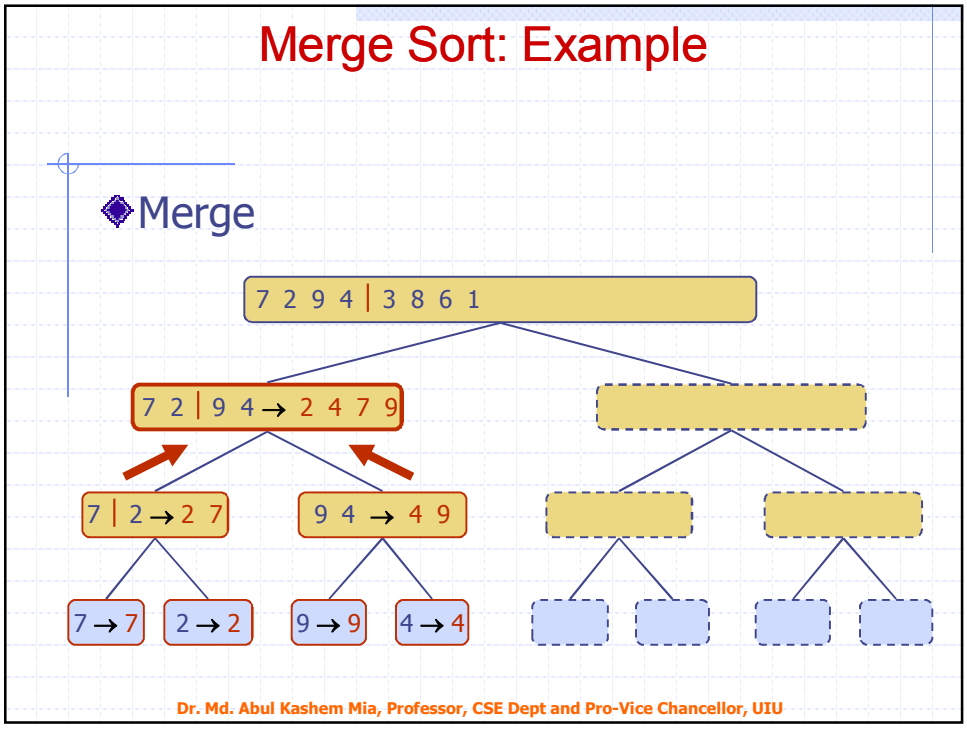

Merge Sort: Example
Recursive call, partition
7 2 9 4 | 3 8 6 1
7 2 | 9 4
7 | 2
Dr. Md. Abul Kashem Mia, Professor, CSE Dept and Pro-Vice Chancellor, UIU

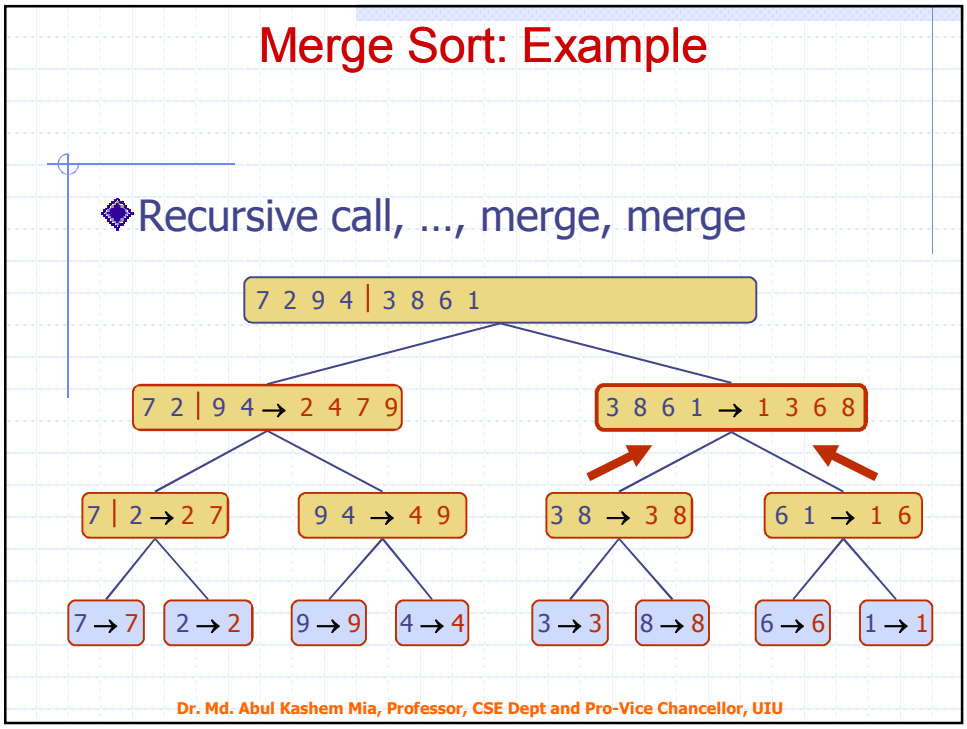

Merge Sort: Example
Recursive call, base case
7 2 9 4 | 3 8 6 1
7 2 | 9 4
7 | 2
7 → 7
Dr. Md. Abul Kashem Mia, Professor, CSE Dept and Pro-Vice Chancellor, UIU

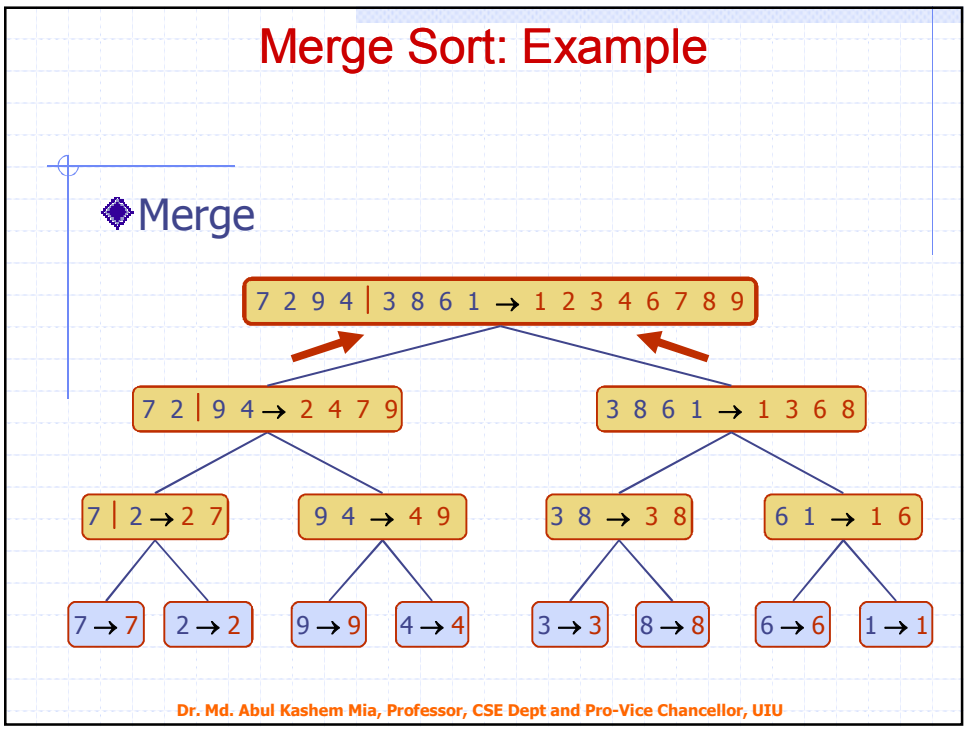

Merge Sort: Example
Recursive call, base case
7 2 9 4 | 3 8 6 1
7 2 | 9 4
7 | 2
7 → 7
2 → 2
Dr. Md. Abul Kashem Mia, Professor, CSE Dept and Pro-Vice Chancellor, UIU


Merge Sort: Example
Merge
7 2 9 4 | 3 8 6 1
7 2 | 9 4
7 | 2 → 2 7
7 → 7
2 → 2
Dr. Md. Abul Kashem Mia, Professor, CSE Dept and Pro-Vice Chancellor, UIU


Merge Sort: Example
Recursive call, ..., base case, merge
7 2 9 4 | 3 8 6 1
7 2 | 9 4
7 | 2 → 2 7
9 4 → 4 9
7 → 7
2 → 2
9 → 9
4 → 4
Dr. Md. Abul Kashem Mia, Professor, CSE Dept and Pro-Vice Chancellor, UIU


Merge Sort: Example
Merge
7 2 9 4 | 3 8 6 1
7 2 | 9 4 → 2 4 7 9
7 | 2 → 2 7
9 4 → 4 9
7 → 7
2 → 2
9 → 9
4 → 4
Dr. Md. Abul Kashem Mia, Professor, CSE Dept and Pro-Vice Chancellor, UIU


Merge Sort: Example
Recursive call, ..., merge, merge
7 2 9 4 | 3 8 6 1
7 2 | 9 4 → 2 4 7 9
3 8 6 1 → 1 3 6 8
7 | 2 → 2 7
9 4 → 4 9
3 8 → 3 8
6 1 → 1 6
7 → 7
2 → 2
9 → 9
4 → 4
3 → 3
8 → 8
6 → 6
1 → 1
Dr. Md. Abul Kashem Mia, Professor, CSE Dept and Pro-Vice Chancellor, UIU


Merge Sort: Example
Merge
7 2 9 4 | 3 8 6 1 → 1 2 3 4 6 7 8 9
7 2 | 9 4 → 2 4 7 9
3 8 6 1 → 1 3 6 8
7 | 2 → 2 7
9 4 → 4 9
3 8 → 3 8
6 1 → 1 6
7 → 7
2 → 2
9 → 9
4 → 4
3 → 3
8 → 8
6 → 6
1 → 1
Dr. Md. Abul Kashem Mia, Professor, CSE Dept and Pro-Vice Chancellor, UIU

Merge: Example