Potential Topic Areas

Games and Graphics

Procedural terrain generation

Optimizing something

Particle systems

2D to 3D model generation

Text Processing

File parsing

Computer Architecture

RISC-V

More Theoretical Topics

Automated Theorem Proving

SAT solvers

Turing machines

Cybersecurity

Cryptography

Robotics

AI / Machine Learning / Data Science

Playing a game

Algorithmic trading

Web Development

Networking

Software Engineering

Front end

Backend

Cloud computing

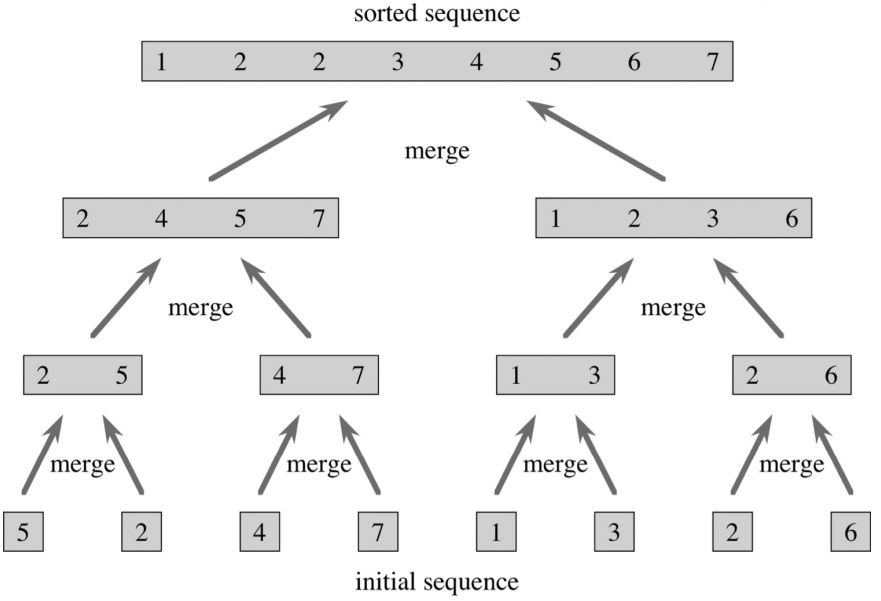
Divide and Conquer

- Recursive approach
- Break a problem into subproblems that are smaller instances of the same problem to create a solution to the original problem

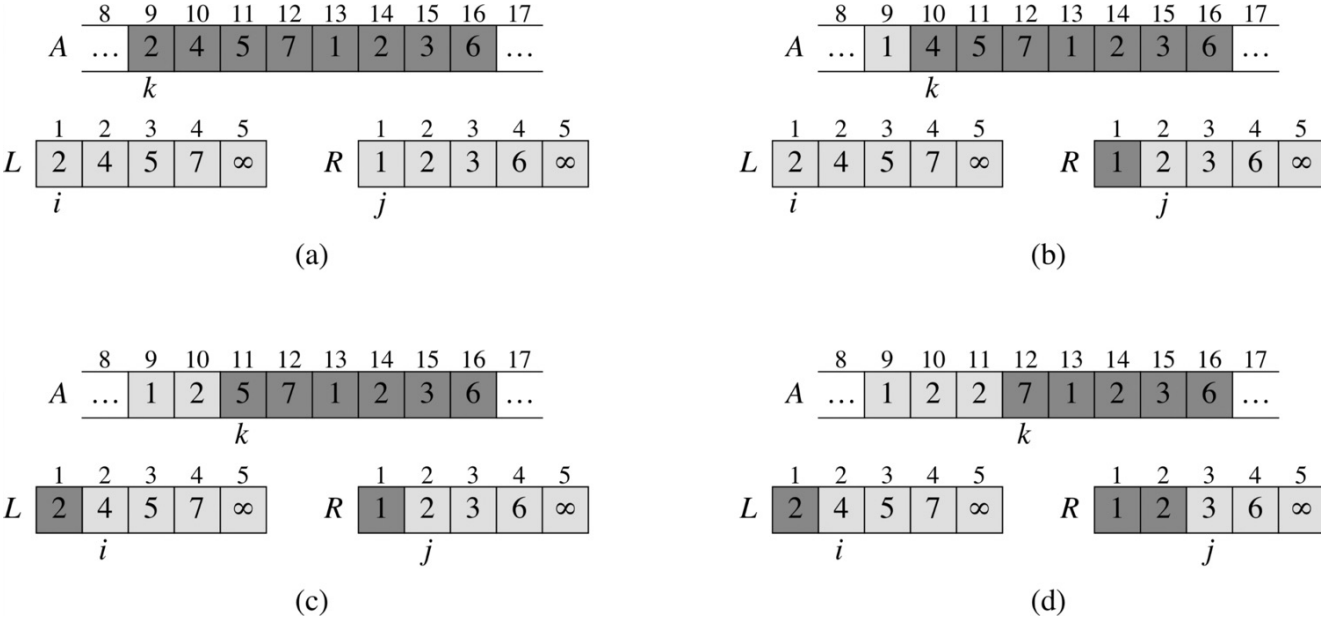
- Steps
  - Divide into subproblems
  - Conquer the subproblems by solving them recursively, or if the subproblem sizes are small enough, solve them in a straightforward manner (base case of recursion)
  - Combine the solutions to subproblems into the solution to the original problem

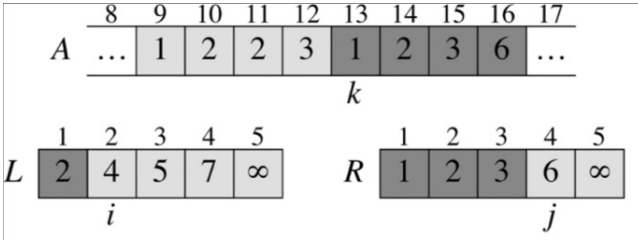
```
MERGE-SORT(A, p, r)
  if p < r                          // check for base case
      q = ⌊(p + r)/2⌋               // divide
      MERGE-SORT(A, p, q)           // conquer
      MERGE-SORT(A, q + 1, r)       // conquer
      MERGE(A, p, q, r)             // combine
```

Sorting subarray $A[p..r]$

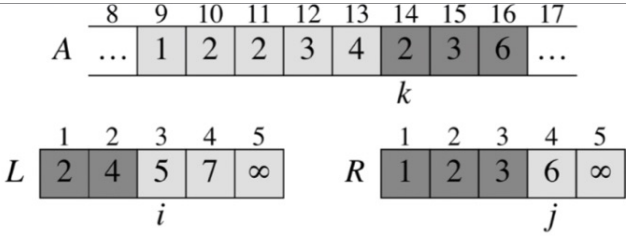

sorted sequence
1 2 2 3 4 5 6 7
merge
2 4 5 7
1 2 3 6
merge
merge
2 5
4 7
1 3
2 6
merge
merge
merge
merge
5
2
4
7
1
3
2
6
initial sequence

Merge sort requires using extra storage during the merge step

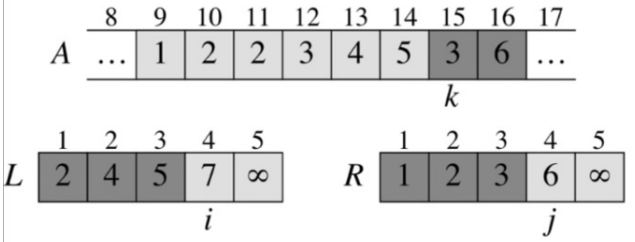



8 9 10 11 12 13 14 15 16 17
A … 1 2 2 3 1 2 3 6 …
k
1 2 3 4 5
L 2 4 5 7 ∞
i
1 2 3 4 5
R 1 2 3 6 ∞
j

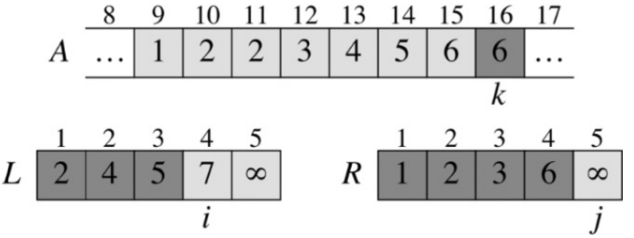
(e)


8 9 10 11 12 13 14 15 16 17
A … 1 2 2 3 4 2 3 6 …
k
1 2 3 4 5
L 2 4 5 7 ∞
i
1 2 3 4 5
R 1 2 3 6 ∞
j

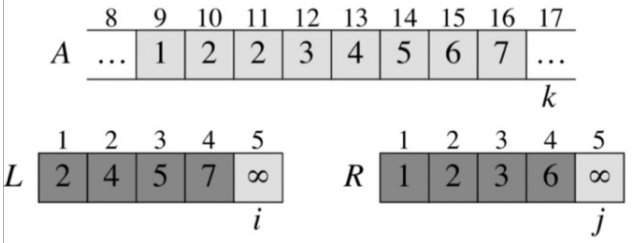
(f)


8 9 10 11 12 13 14 15 16 17
A … 1 2 2 3 4 5 3 6 …
k
1 2 3 4 5
L 2 4 5 7 ∞
i
1 2 3 4 5
R 1 2 3 6 ∞
j

(g)


8 9 10 11 12 13 14 15 16 17
A … 1 2 2 3 4 5 6 6 …
k
1 2 3 4 5
L 2 4 5 7 ∞
i
1 2 3 4 5
R 1 2 3 6 ∞
j

(h)

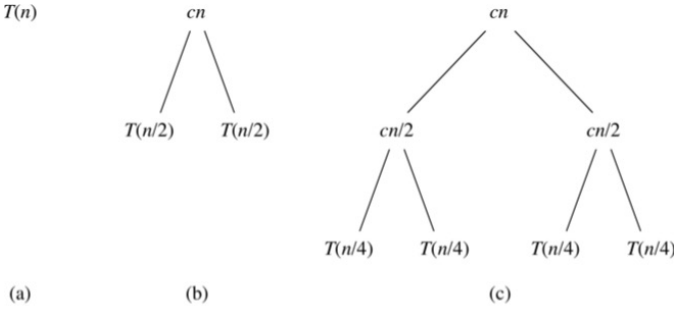

8 9 10 11 12 13 14 15 16 17
A … 1 2 2 3 4 5 6 7 …
k
1 2 3 4 5
L 2 4 5 7 ∞
i
1 2 3 4 5
R 1 2 3 6 ∞
j

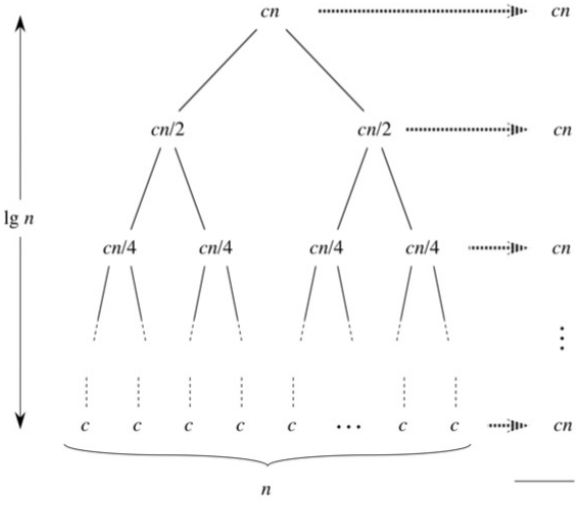
(i)

MERGE(A, p, q, r) Merges the sorted subarrays $A[p..q]$ and $A[q+1..r]$

$n_1 = q - p + 1$
$n_2 = r - q$
let $L[1..n_1 + 1]$ and $R[1..n_2 + 1]$ be new arrays $\Theta(n)$
for $i = 1$ **to** n_1
    $L[i] = A[p + i - 1]$ $\Theta(n)$
for $j = 1$ **to** n_2
    $R[j] = A[q + j]$ $\Theta(n)$
$L[n_1 + 1] = \infty$
$R[n_2 + 1] = \infty$
$i = 1$
$j = 1$
for $k = p$ **to** r $\Theta(n)$
    if $L[i] \le R[j]$
        $A[k] = L[i]$
        $i = i + 1$
    else $A[k] = R[j]$
        $j = j + 1$

Merging is $\Theta(n)$


T(n)
cn
T(n/2)
T(n/2)
cn
cn/2
cn/2
T(n/4)
T(n/4)
T(n/4)
T(n/4)
(a)
(b)
(c)


cn
cn
cn/2
cn/2
cn
lg n
cn/4
cn/4
cn/4
cn/4
cn
c
c
c
c
c
c
c
cn
n
(d)
Total: cn lg n + cn