CSG2A3
ALGORITMA dan STRUKTUR DATA

Single Linked List

Insertion and Deletion

Inserting new Element

› **Insert first**

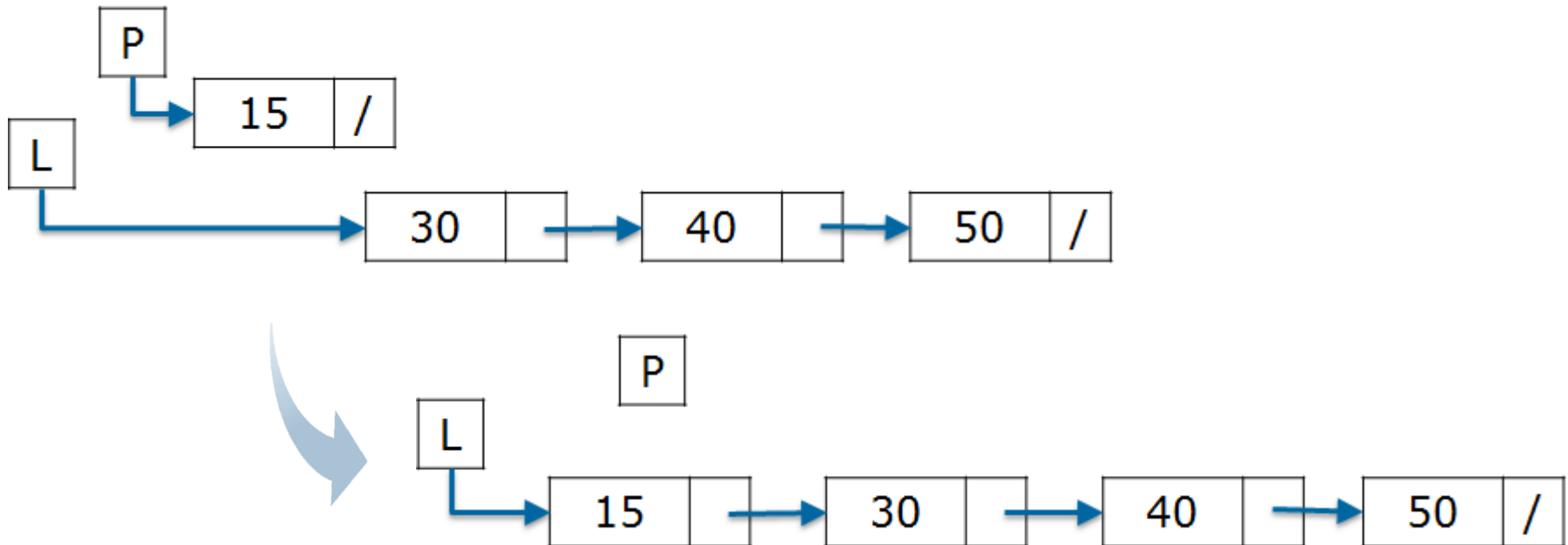– New element became the first element of the list

› **Insert last**

– New element became the last element of the list

› **Insert after / Insert before**

– Put the element somewhere in the middle

Insert First

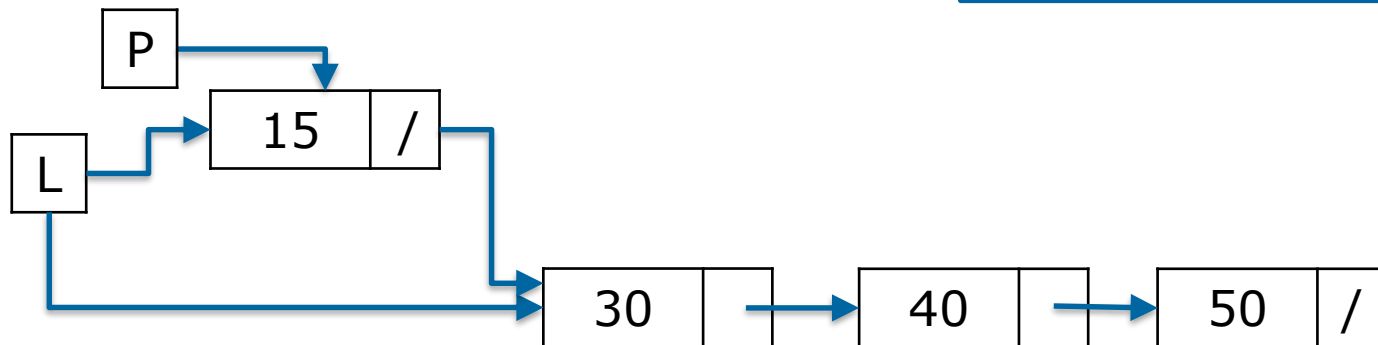› Insert element P into List L so that P become the first element of L
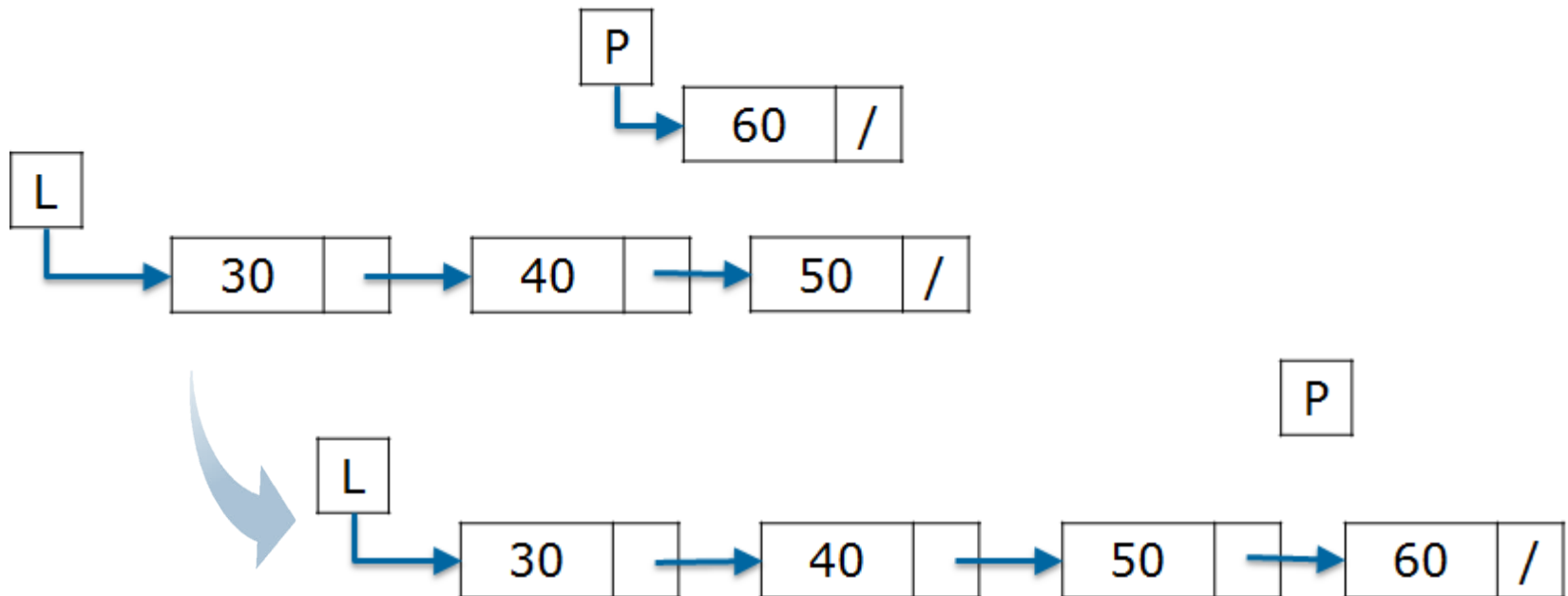
Insert First

Algorithm
> next(P) ← first(L)
>
> first(L) ← P

P

15 | /

L

30 | → 40 | → 50 | /

Insert Last

› Insert element P into List L so that P become the last element of L

Insert Last

Dictionary
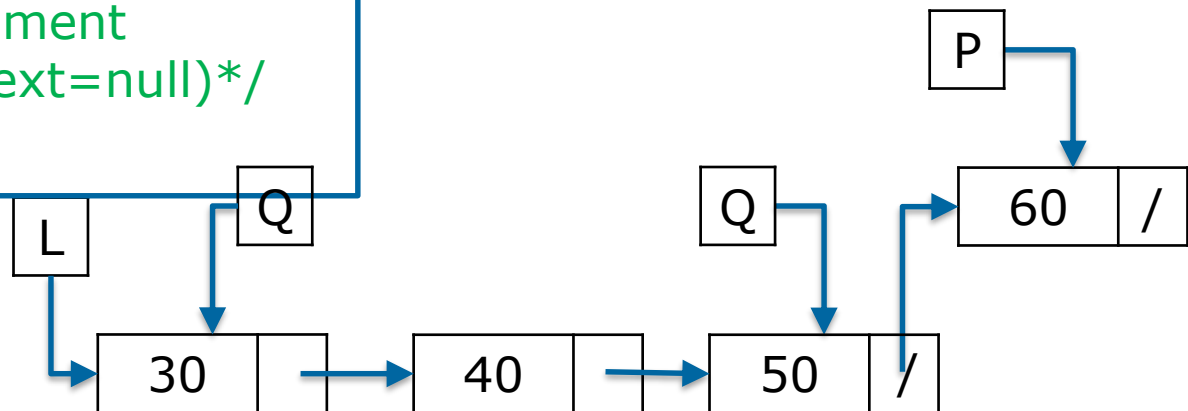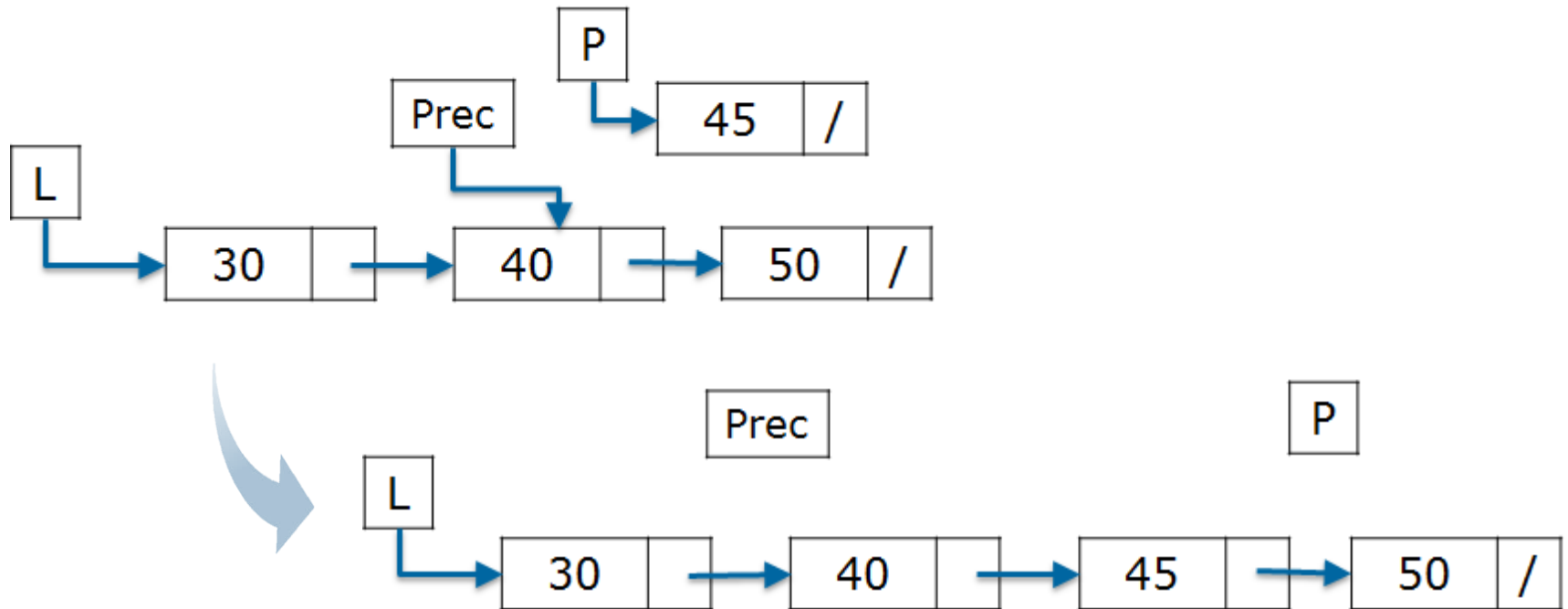
    Q : address

Algorithm

    Q ← first(L)

    /* create a mechanism so that Q

       points the last element

(gunakan loop sampai next=null)*/

    next(Q) ← P

Insert After

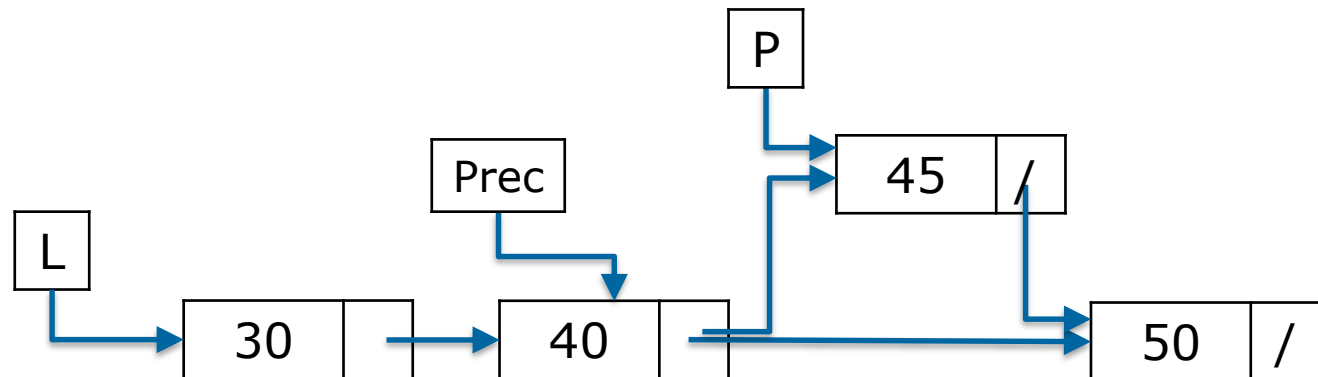› Insert element P into List L so that P become the next element of Prec

Insert After

Algorithm

next(P) ← next(Prec)

next(Prec) ← P

Deleting the Element

› **Delete first**
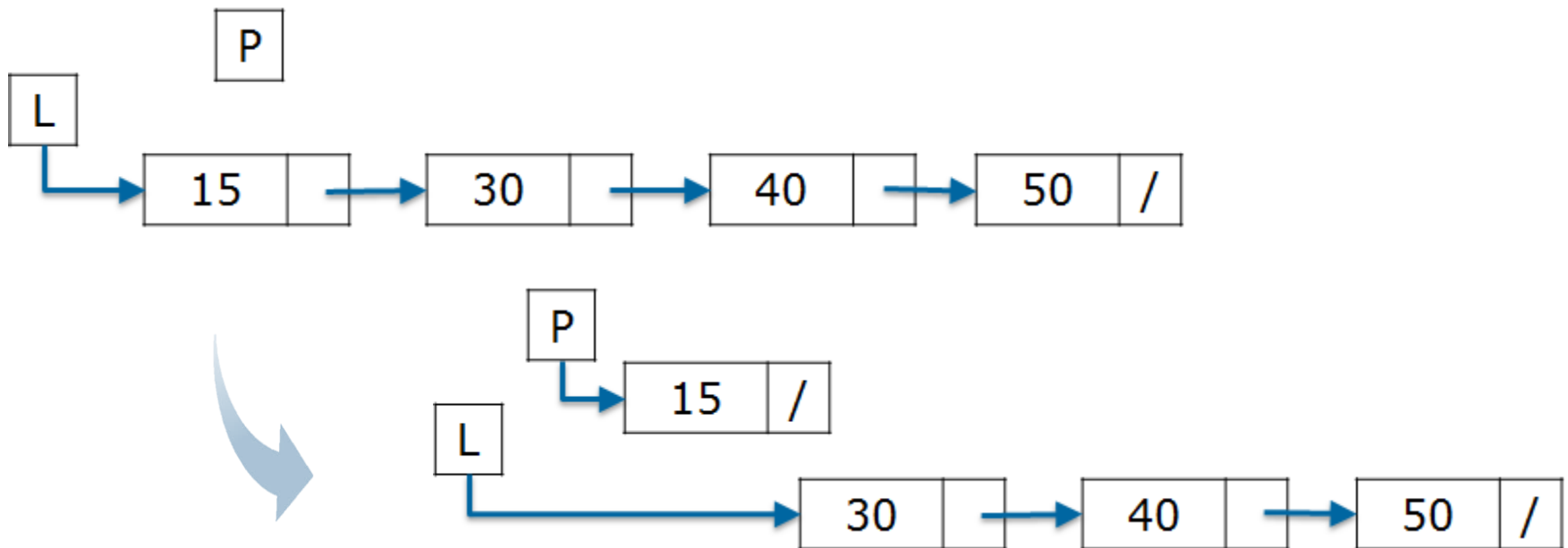 – Remove the first element of the list

› **Delete last**
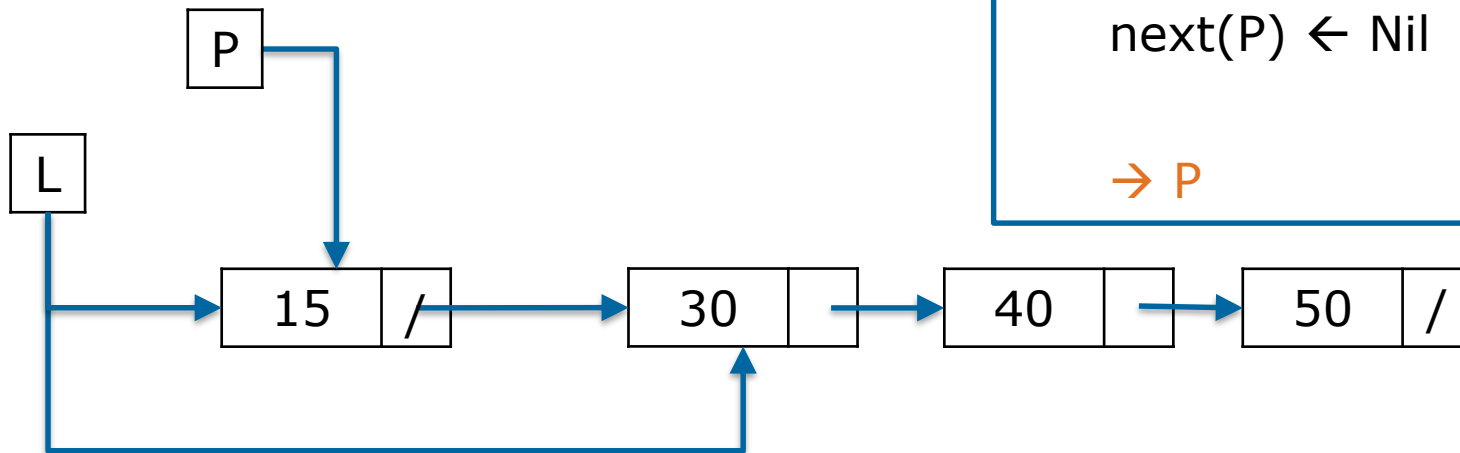 – Remove the last element of the list

› **Delete after**
 – Remove an element next to a particular element

Delete First
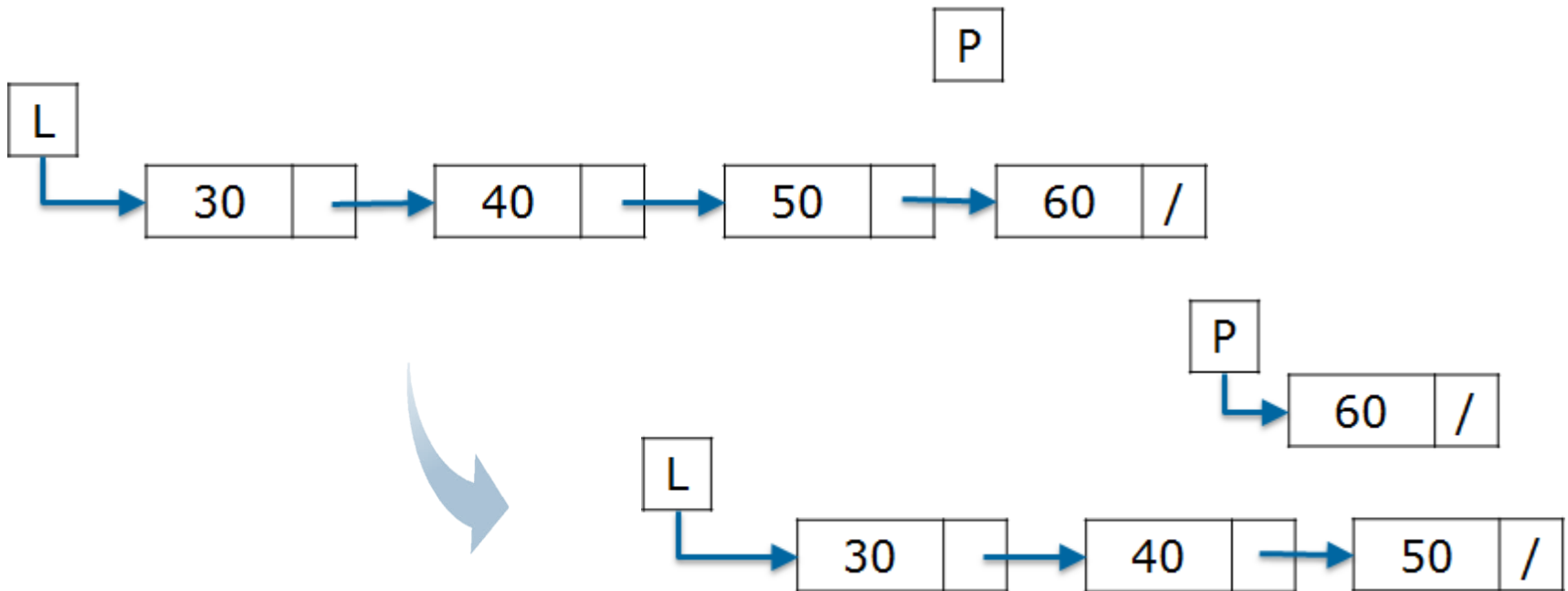
› Remove the first element of L

Delete First

Algorithm

$P \leftarrow first(L)$

$first(L) \leftarrow next(P)$

$next(P) \leftarrow Nil$

$\rightarrow P$

P

L

| 15 | / | | 30 | | | 40 | | | 50 | / |

Delete Last

› Remove the last element of L

Delete Last

Dictionary
    Q : address
Algorithm
    Q ← first(L)
    /* create a mechanism so that Q points
        the element **before** the last element */
    P ← next(Q)
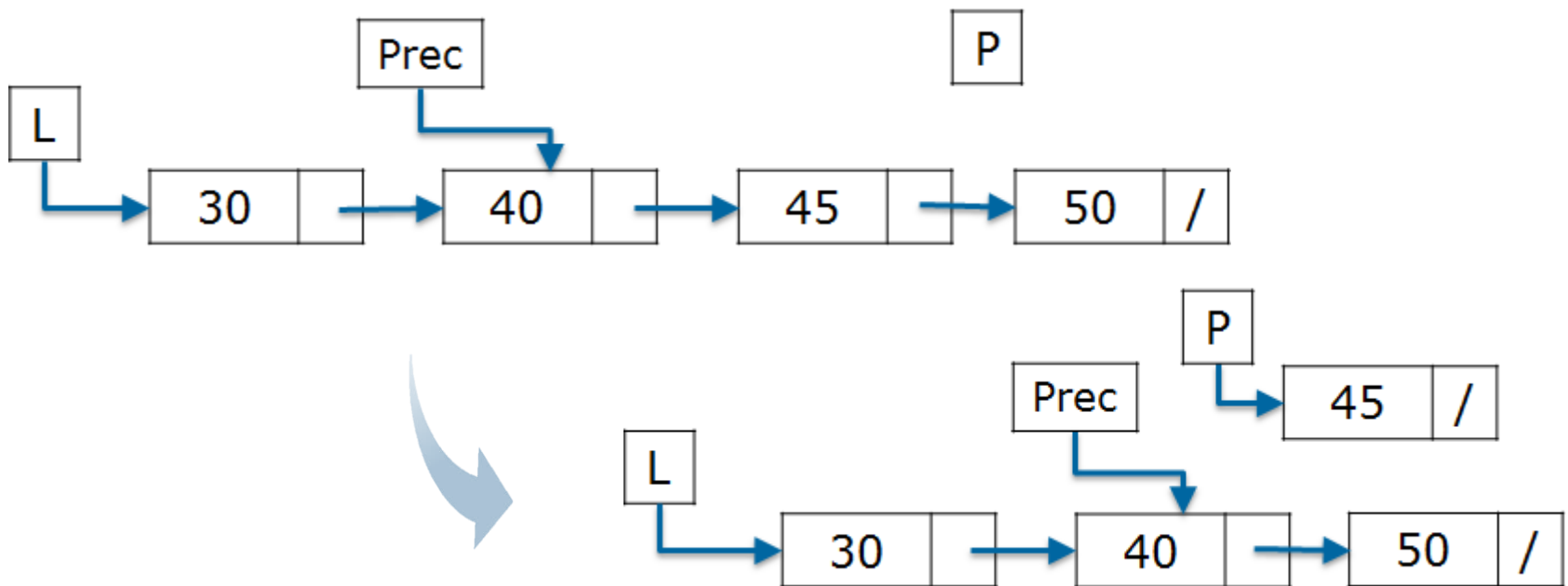   next(Q) ← Nil

    → P
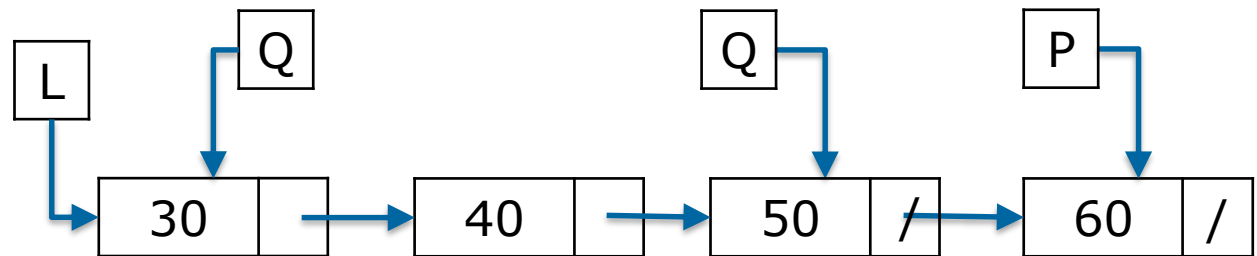
Delete After

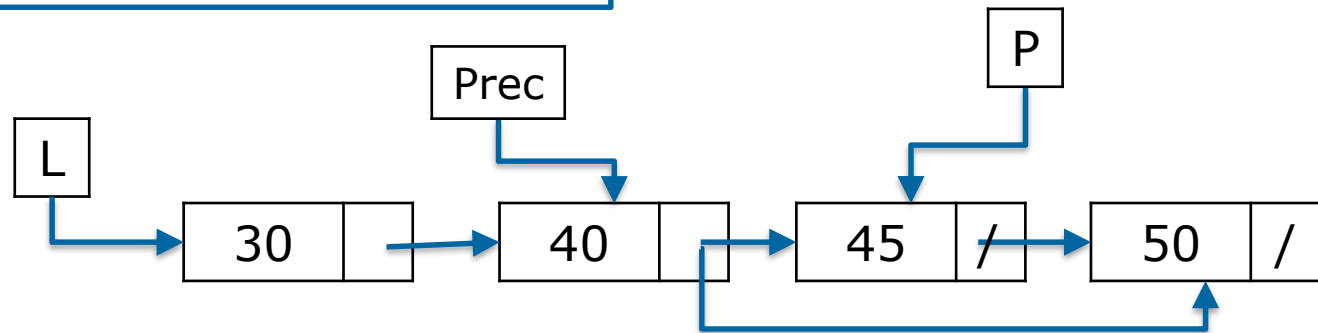› Remove element after the element pointed by Prec

Delete After

Algorithm

    P ← next (Prec)

    next(Prec) ← next(P)

    next(P) ← Nil
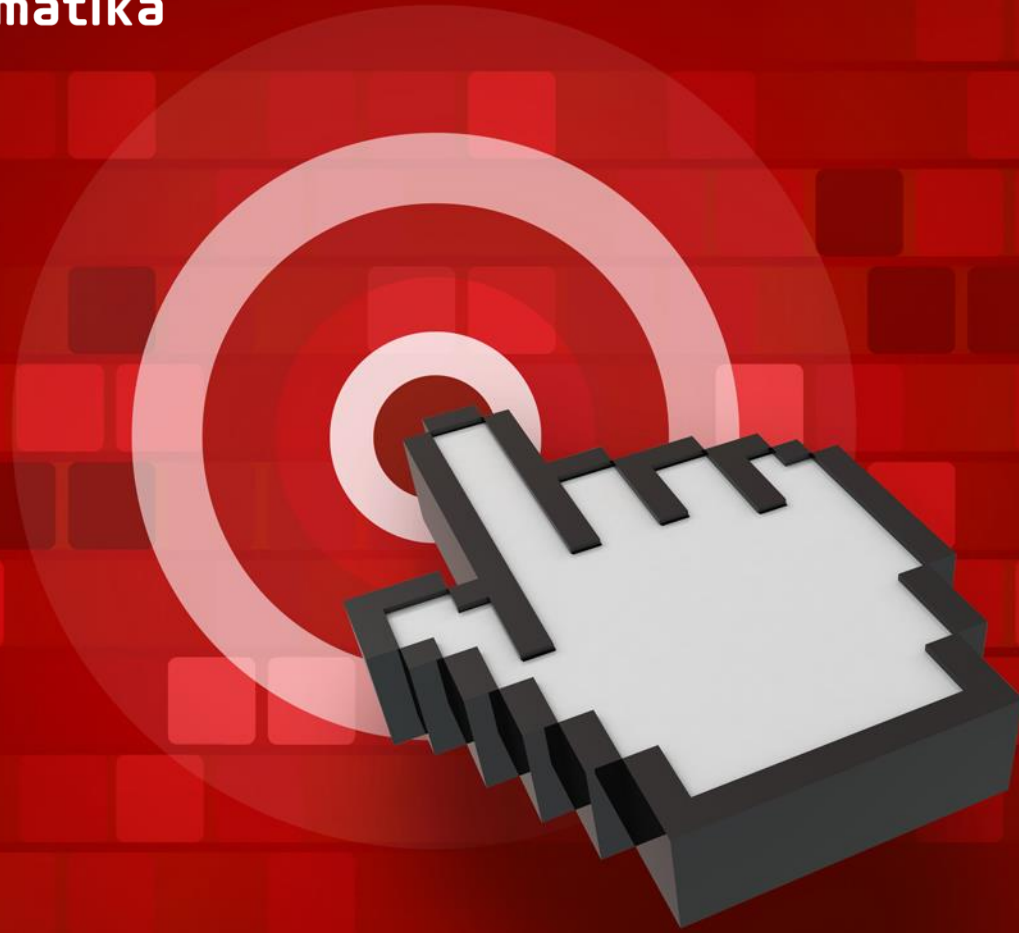

    → P

Mind the special conditions

› Empty list

› Only 1 element in list

Fakultas Informatika
School of Computing
Telkom University

THANK YOU