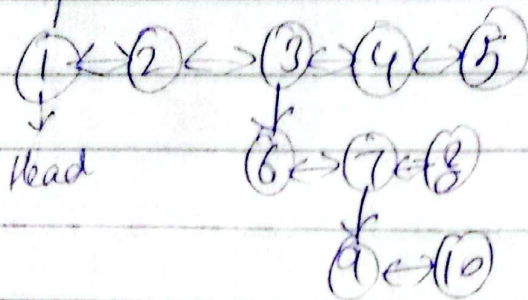


Flatten a Doubly Linked List

Multilevel DLL \rightarrow Node $\begin{cases} \text{next} \\ \text{prev} \\ \text{child} \end{cases}$



Flatten It \rightarrow

Linear

$1 \leftrightarrow 2 \leftrightarrow 3 \leftrightarrow 6 \leftrightarrow 7 \leftrightarrow 9 \leftrightarrow 10 \leftrightarrow 8 \leftrightarrow 4 \leftrightarrow 5$

Recursion Approach

Node* flatten(Node* head) {
 head = NULL; return

Node* curr = head

while (curr != NULL) {

 if (curr->child != NULL) {

 Node* next = curr->next

 curr->next = flatten(curr->child)

 curr->next->prev = curr

 curr->child = NULL

 while (curr->next != NULL) {

 curr = curr->next

 if (next != NULL) {

 curr->next = next

 next->prev = curr

 }

 curr = curr->next

 }

T.C = $O(n)$

connect with next