

1.Split Linked List in Parts.

Shashank Patel CJ

1BM22CS255

```
/**
 * Definition for singly-linked list.
 * struct ListNode {
 *     int val;
 *     struct ListNode *next;
 * };
 */
/**
 * Note: The returned array must be malloced, assume caller calls free().
 */
struct ListNode** splitListToParts(struct ListNode* head, int k, int*
returnSize) {
    struct ListNode **ans = (struct ListNode **)calloc(1, sizeof(struct
ListNode *) * k);
    struct ListNode *prev;
    int base, len = 0, part = 0;

    for (struct ListNode *temp = head; temp; temp = temp->next) {
        len++;
    }

    base = len / k;

    for (int i = len % k; i > 0; i--) {
```

```

        ans[part] = head;
        part++;

        for (int i = 0; i < (base + 1); i++) {
            prev = head;
            head = head->next;
        }

        prev->next = NULL;
    }

    if (base) {
        for (int i = part; i < k; i++) {

            ans[part] = head;
            part++;

            for (int j = 0; j < base; j++) {
                prev = head;
                head = head->next;
            }

            prev->next = NULL;
        }
    }

    *returnSize = k;

    return ans;
}

```

Output:

Accepted Runtime: 0 ms

• Case 1 • Case 2

Input

head =
[1,2,3,4,5,6,7,8,9,10]

k =
3

Output

[[1,2,3,4], [5,6,7], [8,9,10]]

Expected

[[1,2,3,4], [5,6,7], [8,9,10]]

Accepted Runtime: 0 ms

• Case 1 • Case 2

Input

head =
[1,2,3]

k =
5

Output

[[1], [2], [3], [], []]

Expected

[[1], [2], [3], [], []]

The screenshot displays a code editor interface with a dark theme. On the left, a sidebar shows submission details for a user named ShashankPatelCJ. It indicates the submission is 'Accepted' with a runtime of 0 ms and memory usage of 6.56 MB. Below this is a bar chart showing the distribution of memory usage across different time intervals. The main editor area contains C++ code for splitting a linked list into k parts. A user menu is open on the right side of the editor, showing options like 'My Lists', 'Notebook', 'Submissions', 'Progress', 'Points', 'Session', 'Try New Features', 'Orders', 'My Playgrounds', 'Revert to old version', 'Appearance', and 'Sign Out'.

```
1 /**  
2  * Definition for singly-linked list.  
3  * struct ListNode {  
4  *     int val;  
5  *     struct ListNode *next;  
6  * };  
7  */  
8 /**  
9  * Note: The returned array must be malloced, assuming ListNode is defined in the same file  
10 */  
11 struct ListNode** splitListToParts(struct ListNode* head, int k)  
12 {  
13     struct ListNode **ans = (struct ListNode **) malloc(k * sizeof(struct ListNode *));  
14     int base, len = 0, i = 0;  
15     for (struct ListNode *p = head; p != NULL; p = p->next) len++;  
16     base = len / k;  
17     for (int i = 0; i < k; i++)  
18     {  
19         ans[i] = head;  
20         if (base > 0) head = head->next; else head = NULL;  
21         for (int j = 1; j < base; j++) head = head->next;  
22         if (i < k-1) head->next = NULL;  
23     }  
24     return ans;  
25 }
```