```c
/**
 * 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 *tmp = head; tmp; tmp = tmp->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;
```

```
    }

    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;
}
```

Accepted

Sarvesh Rastogi submitted at Feb 20, 2024 12:35

Editorial Solution

Runtime

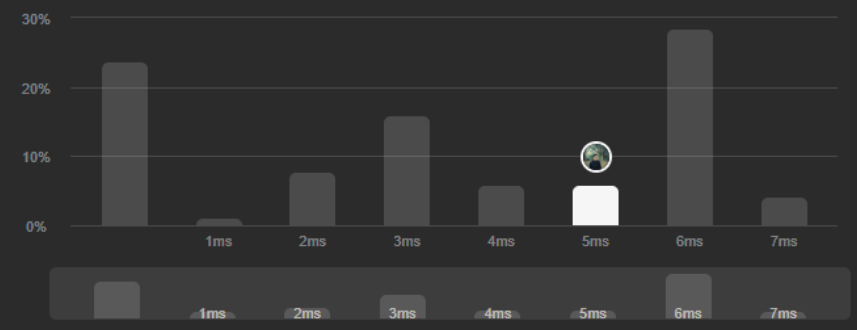
5 ms

Beats 45.56% of users with C

Memory

6.40 MB

Beats 75.74% of users with C



Code | C

```
/**
 * Definition for singly-linked list.
 * struct ListNode {
 *     int val;
 *     struct ListNode *next;
 * };
 */
/**
```

View more