
Started on Thursday, 28 October 2021, 5:45 PM

State Finished

Completed on Thursday, 28 October 2021, 7:11 PM

Time taken 1 hour 26 mins

Marks 320.00/320.00

Grade **100.00** out of 100.00

Time limit	1 s
Memory limit	64 MB

Lengkapi ADT List Berkait dengan representasi fisik pointer yang telah dibuat sebagai tugas pra-praktikum dengan beberapa fungsi/prosedur yang diberikan dalam file [list_linked.h](#) (download-lah dari link yang disediakan), yaitu:

Pencarian sebuah elemen

1. boolean fSearch (List l, Address p);
2. Address searchPrec (List l, ElType x);

Pencarian Nilai Ekstrim dan Rata-Rata (prekondisi: list tidak kosong)

3. ElType max (List l);
4. Address adrMax (List l);
5. ElType min (List l);
6. Address adrMin (List l);
7. float average (List l);

Proses terhadap semua elemen list

8. void delAll (List *l);
9. void inverseList (List *l);
10. List fInverseList (List l);
11. void copyList (List *l1, List *l2);
12. List fCopyList (List l);
13. void cpAllocList (List lIn, List *lOut);
14. void splitList (List *l1, List *l2, List l);

Kumpulkan file [list_linked.c](#).

C

 [list_linked.c](#)

Score: 180

Blackbox

Score: 180

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	10	Accepted	0.00 sec, 1.67 MB
2	10	Accepted	0.00 sec, 1.71 MB
3	10	Accepted	0.00 sec, 1.67 MB
4	10	Accepted	0.00 sec, 1.60 MB
5	10	Accepted	0.00 sec, 1.60 MB
6	10	Accepted	0.00 sec, 1.64 MB
7	10	Accepted	0.00 sec, 1.71 MB
8	10	Accepted	0.00 sec, 1.49 MB
9	10	Accepted	0.00 sec, 1.57 MB
10	10	Accepted	0.00 sec, 1.72 MB
11	10	Accepted	0.00 sec, 1.59 MB
12	10	Accepted	0.00 sec, 1.64 MB

No	Score	Verdict	Description
13	10	Accepted	0.00 sec, 1.71 MB
14	10	Accepted	0.00 sec, 1.51 MB
15	10	Accepted	0.00 sec, 1.71 MB
16	10	Accepted	0.00 sec, 1.67 MB
17	10	Accepted	0.00 sec, 1.57 MB
18	10	Accepted	0.00 sec, 1.65 MB

Question **2**

Correct

Mark 130.00 out of 130.00

Time limit	1 s
Memory limit	64 MB

Pada sistem operasi, akses memori pada RAM cukup lambat. Karena itu, digunakan cache yang jauh lebih cepat. Namun cache berukuran sangat kecil, jadi cache hanya menyimpan sedikit bagian dari RAM. Karena tidak semua nilai disimpan, saat dibutuhkan sebuah nilai x , cache bisa hit (cache memiliki nilai x) atau miss (cache tidak memiliki nilai x).

Salah satu implementasi cache adalah dengan skema LRU, yakni Least Recently Used. Pada skema ini, apabila nilai x tidak ada di cache, cache akan menghapus nilai di cache yang paling lama sudah tidak digunakan. Jadi, cache menyimpan daftar nilai, terurut dari yang paling baru digunakan, sampai yang paling lama digunakan.

Cache dapat direpresentasikan sebagai sebuah list linier. Cache akan diinisialisasi dengan nilai 1 sampai N . Lalu, akan ada Q buah operasi pengambilan nilai x :

- Jika x ada di cache, nilai x dipindah ke depan cache.
- Jika x tidak ada di cache, nilai paling akhir dihapus dari cache dan x dimasukkan ke depan cache.

Untuk setiap operasi, tuliskan apakah operasi "hit" atau "miss". Lalu, tuliskan isi list.

Untuk mengerjakan soal ini, Anda dapat menggunakan header [list_linked.h](#) pada pra praktikum.

Kumpulan **cache.c**

Contoh input / output:

Input	Output	Penjelasan
6		Pada awalnya, cache berisi [1,2,3,4,5,6].
5	hit [4,1,2,3,5,6]	Ada 5 operasi yang dilakukan.
4	hit [6,4,1,2,3,5]	Pada operasi pertama, 4 ada di cache.
6	miss [7,6,4,1,2,3]	Pada operasi kedua, 6 juga ada di cache.
7	miss [5,7,6,4,1,2]	Pada operasi ketiga, 7 tidak ada di cache, jadi 5 dihapus dan 7 ditambahkan ke depan cache.
5	hit [5,7,6,4,1,2]	Pada operasi kelima, 5 sudah tidak ada di cache, jadi 3 dihapus dan 5 ditambahkan ke depan cache.
5		Pada operasi keenam, 5 sudah ada di cache.

C

 [cache.c](#)

Score: 130

Blackbox

Score: 130

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	10	Accepted	0.00 sec, 1.67 MB
2	10	Accepted	0.00 sec, 1.64 MB
3	10	Accepted	0.00 sec, 1.63 MB
4	10	Accepted	0.00 sec, 1.51 MB
5	10	Accepted	0.00 sec, 1.59 MB
6	10	Accepted	0.00 sec, 1.60 MB
7	10	Accepted	0.00 sec, 1.60 MB
8	10	Accepted	0.00 sec, 1.59 MB
9	10	Accepted	0.00 sec, 1.62 MB

No	Score	Verdict	Description
10	10	Accepted	0.00 sec, 1.71 MB
11	10	Accepted	0.00 sec, 1.62 MB
12	10	Accepted	0.00 sec, 1.57 MB
13	10	Accepted	0.00 sec, 1.64 MB

[◀ list_linked.h](#)

Jump to...