

13. Misalnya kita menggunakan Quicksort untuk mengurutkan sebuah array secara ascending. Kemudian, diketahui hasil iterasi pertama dalam algoritma Quicksort menghasilkan array sebagai berikut:

31	14	35	62	61	66	69	68
----	----	----	----	----	----	----	----

Pernyataan yang benar mengenai pivot pada iterasi pertama tersebut adalah:

- A. Data yang mungkin menjadi pivot hanya 35 dan 69.
 - B. Data yang mungkin menjadi pivot adalah 35 dan 66.
 - C. Data yang mungkin menjadi pivot hanya 66.
 - D. Data yang mungkin menjadi pivot adalah 62 dan 68.
14. List di bawah ini merupakan 2 tahap/iterasi sebuah algoritma sorting. Algoritma apakah yang dimaksud?

Data Awal:	134	112	101	105	132	156
Setelah Iterasi 1:	112	134	101	105	132	156
Setelah Iterasi 2:	101	112	134	105	132	156
Setelah Iterasi 3:	105	101	112	134	132	156

- A. Selection Sort iter 1 : 101 112 134 105 132 156
 - B. Bubble Sort iter 1 : 112 101 105 132 134 156
 - C. Insertion Sort**
 - D. Shellsort
15. Suatu algoritma **Quicksort** melakukan partisi dengan algoritma *Partisi-1* (sesuai dijelaskan di materi kuliah) dengan pivot adalah elemen pertama ruas data yang diberikan. Diberikan ruas data awal (sebelum algoritma partisi dijalankan) dan isi ruas data setelah iterasi ketiga (angka yang **bold** adalah pivot):

Ruas Awal:	42	81	35	71	39	41	25	34
Iterasi 1:	34	81	35	71	39	41	25	42
Iterasi 2:	34	42	35	71	39	41	25	81
Iterasi 3:	34	25	35	71	39	41	42	81

Jika iterasi dilanjutkan hingga algoritma partisi itu selesai, tuliskanlah hasil akhir partisi tersebut.

Iterasi 4: 34 25 35 42 39 41 71 81

4689-6

Iterasi 5: 34 25 35 41 39 42 71 81

Isikan bilangan-bilangan pada kotak-kotak kosong di bawah ini sesuai urutan hasil partisi.

Setelah Partisi	34	25	35	41	39	42	71	81
-----------------	----	----	----	----	----	----	----	----

16. Diberikan sebuah array sebagai berikut:

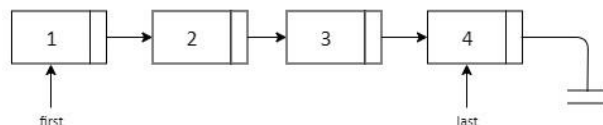
76	32	63	18	25	50	42	14	51	9
9	14	50	18	25	51	42	32	63	76

Elemen akan diurutkan secara ascending (kecil ke besar) dengan menggunakan Shellsort. Terapkan 5-sort (tahapan dari algoritma Shellsort dengan gap = 3), dan tuliskan isi array setelah selesai dilakukan 5-sort tersebut ...

Jawaban (isikan pada kotak-kotak kosong hasil 5-sort tsb):

9	14	50	18	25	51	42	32	63	76
---	----	----	----	----	----	----	----	----	----

Berikut ini adalah deskripsi untuk soal-soal no 17 dan 18. Singly linked-list didefinisikan dengan **class MyNode** untuk node-nya. Constructor class node adalah **MyNode(int value, MyNode next)**, **value** merupakan data yang disimpan, dan **next** menyimpan alamat next node. Node pertama ditunjuk oleh variabel **first** dan node terakhir oleh variabel **last** sebagai digambarkan pada ilustrasi berikut ini.



17. Saat ini, linked list tersebut sudah berisi N buah node ($N > 0$) dan terdapat variabel **current**, yang menunjuk ke salah satu node. Manakah **deret perintah yang salah** untuk menambahkan sekaligus **dua node baru** yang akan berisi **value 10, dan value 20 (berurutan)** pada posisi berikutnya setelah **current**? misal awalnya: **current** → node A

- ✓ A. `current.next = new MyNode(20, current.next);` **current** → 20 → node A
`current.next = new MyNode(10, current.next);` **current** → 10 → 20 → node A
- ✓ B. `current.next = new MyNode(10, current.next);` **current** → 10 → node A
`current.next.next = new MyNode(20, current.next.next);` **current** → 10 → 20 → node A
- ✗ C. `tmp = new MyNode(10, current.next);` tmp : 10 → node A
`current.next = new MyNode(20, tmp);` **current** → 20 → 10 → node A
- ✓ D. `tmp = new MyNode(20, current.next);` tmp : 20 → node A
`current.next = new MyNode(10, tmp);` **current** → 10 → 20 → node A

18. Jika linked list berisi N buah node ($N \gg 100$), tentukan operasi mana yang kompleksitasnya adalah $O(N)$?

- A. Insertion pada posisi sebelum elemen ke-10 dari awal Linked-list. $O(9) \Rightarrow O(1)$
- B. Hapus elemen pada posisi 3 dari belakang (pada posisi $N-2$). $O(N-2) \Rightarrow O(N)$
- C. Remove elemen setelah elemen ke-40. $O(41) \Rightarrow O(1)$
- D. Remove elemen sebelum elemen ke-10 $O(9)$

Berikut ini adalah deskripsi untuk soal-soal no 19 dan 20. Sebuah queue

diimplementasikan dengan circular array berukuran N tertentu (indeks dari 0 s.d. $N-1$).

Variabel **back** menunjuk posisi berikutnya yang akan ditempati data yang baru di-enqueue dan variabel **front** menunjuk posisi berikutnya data yang akan di-dequeue.

19. Suatu saat queue yang berukuran $N=6$ terisi penuh dengan $\text{front}=3$ dan $\text{back}=3$, dan terjadi enqueue data berikutnya sehingga dilakukan array doubling menjadi berukuran

12. Posisi array beserta data di dalamnya sebelum array doubling adalah sbb:

enqueue : add item di back, back geser
 dequeue : remove item di front, front geser

	0	1	2	3	4	5
data	D	E	F	A	B	C

back
front

0 1 2 3 4 5 6 7 8 9 10 11
 A B C D E F X
 f b

Setelah array doubling harga front dan back adalah:

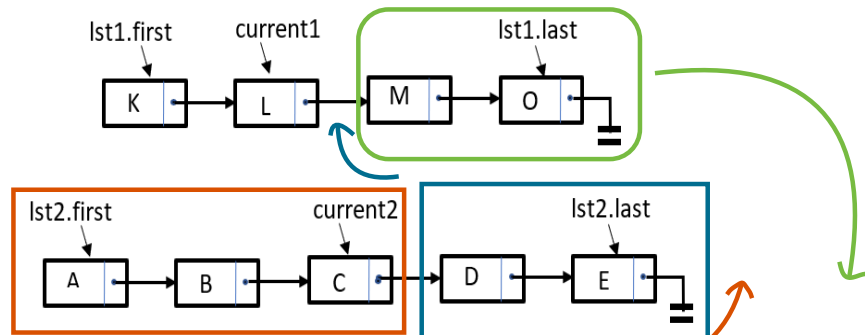
- A. Front = 0 back = 5
- B. Front = 3 back = 5
- C. Front = 5 back = 3
- D. Front = 0 back = 0

20. Jika queue berukuran $N=5$ sel mula-mula kosong dengan **back** dan **front** berharga 0).

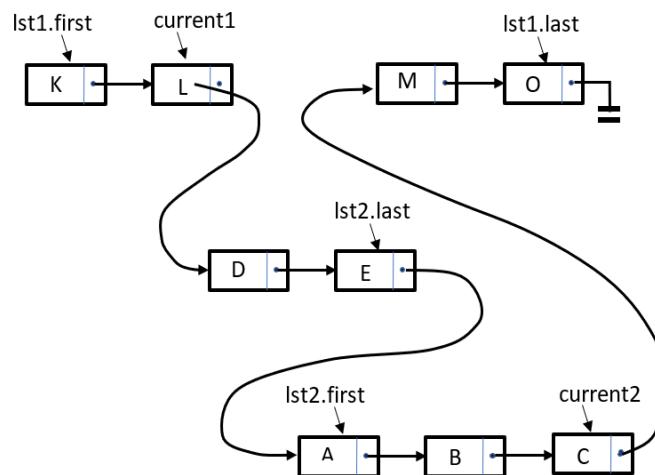
Pada queue dilakukan sederetan operasi enqueue/dequeue sebagai berikut:

- 1) enqueue "kucing"
- 2) enqueue "kera"
- 3) enqueue "anjing"
- 4) dequeue
- 5) dequeue
- 6) enqueue "ayam"
- 7) enqueue "angsa"
- 8) dequeue
- 9) enqueue "kelinci"

3. Diberikan dua buah linked list: **lst1** dan **lst2**. Masing-masing memiliki pointer **first** yang menunjuk elemen pertama dan **last** yang menunjuk elemen terakhir pada masing-masing linked list. Suatu saat kedua linked list tidak kosong dan terdapat **current1** yang menunjuk suatu node dalam lst1 dan **current2** pada suatu node dalam lst2 (Lihat Illustirasi pada gambar berikut).



Node-node **setelah current2** hingga **lst2.last** akan ditempatkan **setelah current1** dari lst1, dan node-node **lst2.first** hingga **current2** akan ditempatkan di antara **lst2.last** dan setelah current1 semula (lihat ilustrasi pada gambar berikut).



Tuliskan beberapa baris perintah (**hints**: hanya sekitar 4 baris perintah!) untuk melakukan hal tersebut agar operasi dapat dilakukan sekaligus ($O(1)$).

<code>temp = current1.next</code>	<code>temp : M O</code>
<code>current1.next = current2.next</code>	<code>K L D E</code>
<code>lst2.last.next = lst2.first</code>	<code>K L D E A B C (D E)</code>
<code>current2.next = temp</code>	<code>K L D E A B C M O</code>
<code>temp = current2.next</code>	<code>temp : D E</code>
<code>current2.next = current1.next</code>	<code>A B C M O</code>
<code>current1.next = temp</code>	<code>K L D E</code>
<code>lst2.last.next = lst2.first</code>	<code>K L D E A B C M O</code>