LinkedList, Stack dan Queue

Bagian A
Isian Singkat (15 Soal)

1. Sebuah linked list menggunakan object Node sebagai nodenya. Node pertama dicatat dengan variabel first. Linkedlist **TIDAK menggunakan dummy-header** (sehingga elemen yang ditunjuk first berisi actual data). Anda akan menghapus node pada posisi current. Method prev(current) disediakan untuk mendapatkan node sebelum posisi current. Prev(current) akan me-return null jika current berada pada posisi pertama linkedlist. Seseorang telah menuliskan beberapa baris code untuk itu, sbb.
   ```
   prev(current).next = current; current=current.next;
   ```
   Apakah code tsb sudah benar?
   Jika sudah benar jawaban anda adalah “benar”. Jika belum, tuliskan perintah-perintah yang seharusnya.

2. Apa *yang menyebabkan overhead cost* (biaya tambahan) dalam penggunaan struktur LinkedList?

3. Sebuah linked list menggunakan object Node sebagai nodenya. Node pertama dicatat dengan variabel first. Linkedlist **TIDAK menggunakan dummy-header** (sehingga elemen yang ditunjuk first berisi actual data). Constructor Node(val,addr) akan mengcreate node Node dan menyimpan data val dalam node serta mengisi alamat next dalam Node dengan harga addr. Anda akan menyisipkan node baru berharga X setelah node yang ditunjukkan oleh variabel current. Seseorang telah menuliskan beberapa baris code untuk itu, sbb.
   ```
   Node newnode = new Node(val, current.next);  // baris 1
   current.next = current;  // baris 2
   ```
   Baris mana yang harus diperbaiki dan seharusnya tertulis apa?

4. Sebuah linked list menggunakan object Node sebagai nodenya. Node pertama dicatat dengan variabel first. Linkedlist **TIDAK menggunakan dummy-header** (sehingga elemen yang ditunjuk first berisi actual data). Constructor Node(val,addr) akan mengcreate node Node dan menyimpan data val dalam node serta mengisi alamat next dalam Node dengan harga addr. Anda akan menyisipkan node baru berharga val **sebagai elemen pertama linked list**.
   ```
   a) first = new Node(val, null); first = first.next;
   b) Node newnode = new Node(val, first); first = newnode;
   c) first = new Node(val, first); Node newnode = first;
   d) Node newnode = new Node(val, first); newnode = first;
   e) Node newnode = first;  new Node(var, newnode); first = newnode;
   f) Node newnode = first; first = new Node(val, first);
   g) first = first.next; first = new Node(val, null);
   ```
   Manakah deretan perintah yang benar? Jawab dengan menuliskan huruf-huruf di depan pilihan tsb dan jika lebih dari satu, pisahkan dengan satu spasi dan terurut.

5. Manakah di antara struktur data linear berikut yang memiliki growth rate terbaik pada penambahan data?
   1. unsorted array (asumsi space mencukupi)
   2. sorted array (asumsi space mencukupi)
   3. sorted linked list tanpa menyimpan last
   4. unsorted linked list

   Tuliskan jawaban anda dengan nomor di depan pilihan di atas. Jika lebih dari satu pisahkan dengan spasi.

6. Sebuah linked list menggunakan object Node sebagai nodenya. Node pertama dicatat dengan variabel first. Linkedlist **menggunakan dummy-header** (sehingga elemen yang ditunjuk first adalah DUMMY node). Anda akan mencari node terakhir (node terakhir tidak disimpan secara

khusus). Jika linkedlist kosong (hanya berisi dummy node) maka harus mereturn null. Seseorang telah menuliskan beberapa baris code untuk itu, sbb.

```
Node searchnode = first;       //baris1
while (searchnode.next != null) //baris2
searchnode = searchnode.next; //baris3
return searchnode;             //baris4
```

Apakah code tsb sudah benar untuk semua kasus?
Jika sudah benar untuk semua kasus tulis jawaban anda dengan kata “benar”. Jika belum, tulis jawaban anda dengan kata “salah, ….” Dengan menuliskan kasus apa yang bermasalah itu (kalau lebih dari satu, sebutkan salah satu saja) pada bagian “….”.

7. Apkah kompleksitas waktu untuk membalikkan urutan isi linked list berukuran N ?

8. Jika sorting dilakukan pada Linked List (tanpa penggunaan temporary storage array) berukuran N akan memerlukan waktu dengan kompleksitas apakah?

9. Sebuah linked list menggunakan object Node sebagai nodenya. Node pertama dicatat dengan variabel first. Linkedlist **TIDAK menggunakan dummy-header** (sehingga elemen yang ditunjuk first berisi actual data). Method Node.sameval(val) akan mereturn true jika harga data dalam node ybs sama dengan harga val, atau false jika tidak. Anda akan mencari suatu node berharga val dan me-return node yang berisi harga tersebut. Seseorang telah menuliskan potongan program sbb. untuk itu.

```
Node searchnode = first;       //baris1
while (searchnode != null) {   //baris2
if (searchnode.sameval(val))   //baris3
return searchnode;             //baris4
}                              //baris5
return null;                   //baris6
```

Apakah potongan program itu sudah benar?
Jika sudah benar jawaban anda adalah “benar”. Jika belum benar, tuliskan apa yang harus diubah/ditambahkan dan pada/setelah baris mana.

10. Apa manfaat dari penggunaan dummy header node pada linked list?

11. Bandingkan kompleksitas waktu LinkedList dan sorted-array dalam menemukan suatu data di antara N data, dan tuliskan jawaban dalam notasi Big-O dari kedua algoritma tersebut (urutan sesuai di atas).

12. Keuntungan doubly linked list dibandingkan singly linked list adalah dalam menemukan....

13. Sebuah linked list menggunakan object Node sebagai nodenya. Node pertama dicatat dengan variabel first. Linkedlist **TIDAK menggunakan dummy-header** (sehingga elemen yang ditunjuk first berisi actual data). Constructor Node(val, addr) akan mengcreate node Node dan menyimpan data val dalam node serta mengisi alamat next dalam Node dengan harga addr. Seseorang telah menuliskan beberapa baris code untuk itu, sbb.

```
Node newnode = new Node(current.data, current.next);
current.data = val;
current.next = newnode;
```

Efek apakah yang terjadi (ditinjau dari isi datanya)?
Jawaban anda dituliskan sependek-pendeknya, misalnya “menghapus data current”, “menghapus data sebelum current”, “menyisipkan data val setelah current”, menyisipkan data val sebelum current”, “mengganti isi current dengan val”, dsb.

14. Sebutkan salah SATU keuntungan paling penting dari penggunaan Linked List dibandingkan array (jawaban anda berisi maksimum 7 kata!).

15. Sebuah doubly linked list menggunakan object DoublyNode sebagai nodenya dengan prev menunjuk ke sebelum dan next menunjuk ke setelah. Anda akan menghapus node pada posisi current. Asumsi bahwa current berada di tengah dua node yang benar-benar ada.
    1. `current.next.prev = current.prev; current.prev.next = current.next;`
    2. `current.next.prev = current.next; current.prev.next = current.prev;`
    3. `current.next.prev = current; current.prev.next = current;`
    4. `current.next = current.prev.next; current.prev = current.next.prev;`
    5. `current.next.prev = current.prev.next; current.prev.next= current.next.prev;`
    6. `current.prev.next = current.next; current.prev.next.prev = current.prev;`

    Manakah deretan perintah yang benar?
    Jawab dengan menuliskan angka-angka di depan pilihan tersebut dan jika lebih dari satu, pisahkan dengan satu spasi dan terurut membesar.

16. Pada implementasi Stack dengan array, tanpa memperhatikan array-doubling, berapa banyaknya variabel (minimal) yang perlu digunakan untuk pendukung bekerjanya stack itu?

17. Pada implementasi Queue dengan array, tanpa memperhatikan array-doubling, berapa banyaknya variabel (minimal) yang perlu digunakan untuk pendukung bekerjanya queue itu?

18. Imlementasi Queue tanpa adanya konsep circular array, maka akan muncul masalah penggeseran isi queue, yaitu ketika kondisi .....

19. Pada implementasi stack dengan array, terdapat sejumlah masalah yang muncul (dibandingkan dengan linked list). Manakah dari berikut ini YANG BUKAN masalah tersebut dan jawablah dengan menuliskan huruf (huruf-huruf) di depannya secara terurut dipisahkan spasi.
    a) Banyaknya memory space yang dialokasi lebih banyak dari kenyataan jumlah data yang ditangani
    b) Perlu adanya operasi array doubling O(n) saat stack penuh dan ada data berikutnya yang akan masuk dalam stack
    c) Operasi mendapatkan top-of-stack adalah O(n) karena perlu pencarian sikuensial
    d) Perlu pemeriksaan kondisi stack kosong (saat operasi pop) atau penuh (saat operasi push)

20. Dalam implementasi queue dengan circular array, variable front menunjukkan posisi elemen berikutnya yang akan di dequeue, dan back menunjukkan posisi kosong berikutnya yang akan ditempati saat enqueue. Jika ditemukan kondisi front=back, ada dua kondisi yang bisa terjadi, kondisi-kondisi queue apa sajakah itu? Jawab dengan singkat dalam satu baris (maksimum 7 kata).

21. Suatu stack diimplementasikan dengan linkedlist. Jika push() dan pop() terjadi di bagian tail maka operas-operasi itu memiliki kompleksitas .....[untuk push]..... dan .....[untuk pop]..... (Tuliskan jawaban anda dalam notasi big-O dengan N banyaknya data dalam stack).

22. Suatu stack diimplementasikan dengan linkedlist. Jika push() dan pop() terjadi di bagian head maka operas-operasi itu memiliki kompleksitas .....[untuk push]..... dan .....[untuk pop]..... (Tuliskan jawaban anda dalam notasi big-O dengan N banyaknya data dalam stack).

23. Suatu queue diimplementasikan dengan linkedlist. Jika enqueue() terjadi di bagian head dan dequeue() terjadi di bagian tail maka operasi-operasi itu memiliki kompleksitas .....[untuk

enqueue]..... dan .....[untuk dequeue]..... (Tuliskan jawaban anda dalam notasi big-O dengan N banyaknya data dalam queue).

24. Suatu queue diimplementasikan dengan linkedlist. Jika enqueue() terjadi di bagian tail dan dequeue() terjadi di bagian head maka operasi-operasi itu memiliki kompleksitas .....[untuk enqueue]..... dan .....[untuk dequeue]..... (Tuliskan jawaban anda dalam notasi big-O dengan N banyaknya data dalam queue).

25. Pada implementasi queue dengan circular array, data berada pada array dataQueue, banyak data dalam queue dicatat pada variabel n, back mencatat posisi kosong berikutnya yang akan dienqueue, front mencatat posisi data berikutnya yang akan didequeue. Saat queue penuh dilakukan dengan array doublig sbb. Seseorang telah menulis code untuk melakukan array doubling dataQueue sebagai berikut.

```
Benda[] tmp = dataQueue;
dataQueue = new Benda[tmp.length * 2];
for (int j = 0; j < tmp.length; j++)
    dataQueue[j] = tmp[j];
```

Pernyataan manakah yang benar setelah perintah-perintah itu dijalankan?

1. Sudah selesai (tidak ada operasi lainnya)
2. gagal, karena ukuran array baru bukan 2 kali semula
3. belum selesai karena data mulai posisi 0 hingga front-1 harus dipindah ke posisi n dan seterusnya serta mengupdate back += n
4. Belum selesai karena data mulai posisi 0 hingga n-1 harus digeser sejumlah posisi sesuai dengan nilai front sehingga yang semula 0 menempati posisi front dan seterusnya hingga posisi back
5. Belum selesai karena front harus diubah menjadi 0 dan back menjadi n-1
6. Gagal karena ada sejumlah data yang hilang akibat adanya wraparound

Tuliskan jawaban anda berupa angka dari pilihan yang benar.

Bagian B
Pilihan berganda (15 Soal)

1. Suatu *linked-list* didefinisikan menggunakan suatu *class* bernama Node. Dalam Node terdapat field next untuk menunjuk Node berikutnya. Node pertama ditunjuk oleh variable headNode. Jika suatu saat dalam *linked-list* sudah berisi minimal satu Node, suatu node baru yang ditunjuk newNode hendak disisipkan pada posisi node kedua dalam *linked-list*. Maka, manakah deretan perintah yang melakukan hal itu dengan benar:

   A.
   ```
   newNode.nextNode = headNode.nextNode;
   headNode.nextNode = newNode.nextNode;
   ```
   B.
   ```
   headNode.nextNode = newNode.nextNode;
   newNode.nextNode= headNode.nextNode;
   ```
   C.
   ```
   headNode.nextNode = newNode;
   newNode.nextNode = headNode.nextNode;
   ```
   D.
   ```
   newNode.nextNode = headNode.nextNode;
   headNode.nextNode = newNode;
   ```
   E.
   ```
   headNode = newNode.nextNode;
   newNode = headNode.nextNode;
   ```

2. berikut ini merupakan kelebihan dari struktur data linked-list dibandingkan dengan array.
   A. Operasi pencarian umumnya lebih cepat dari array
   B. Operasi penghapusan node umumnya lebih lama dari array
   C. Operasi penyisipan umumnya lebih cepat dari array
   D. Operasi penghapusan di akhir umumnya lebih lama dari array
   E. Operasi penghapusah node pertama lebih cepat dari array

3. Berapakah kompleksitas algoritme binary search jika dipaksakan untuk diterapkan pada doubly-linked list yang berisi n data (i.e. mendapatkan pertengahan, lalu bergerak seperempatnya ke kiri/kanan, lalu bergerak seperdelapannya ke kiri/kanan, dst)?
    A. O(n)
    B. O(log n)
    C. O(n^2)
    D. O(log log n)
    E. O(n log n)

4. Penambahan dummy node di awal pada linked-list (dikenal juga sebagai sentinel) berguna untuk berikut ini, kecuali:
    A. Mengurangi perlunya penanganan khusus kasus linked list kosong.
    B. Mengurangi perlunya penanganan khusus kasus “insert after” (setelah current).
    C. Mengurangi perlunya penanganan khusus kasus first == last == null.
    D. Mengurangi perlunya penanganan khusus kasus “insert first” (penyisipan sebelum node pertama).
    E. Mengurangi perlunya penanganan khusus kasus “delete first” (penghapusan node pertama).

5. Apa yang bisa dilakukan oleh doubly linked-list tetapi tidak oleh singly linked-list adalah:
    A. Mendapatkan node berikutnya secara O(1).
    B. Mendapatkan node pertama secara O(1).
    C. Mendapatkan node sebelumnya secara O(1).
    D. Mendapatkan node terakhir secara O(1).
    E. Mendapatkan node berharga tertentu secara O(1).

6. Setiap operasi push dalam stack dengan implementasi array saat array sudah terisi penuh dan menerima data berikut dengan cara doubling kapasitas memerlukan kompleksitas waktu
    A. O(1)
    B. O(log n)
    C. O(n)
    D. O(n^2)
    E. O(n log n)

7. Keuntungan "yang paling jelas/pasti" dari implementasi array di bandingkan dengan implementasi linked-list untuk stack adalah:
    A. Operasi pop selalu lebih cepat
    B. Ukuran memori space yang digunakan selalu lebih sedikit
    C. Operasi push selalu lebih cepat
    D. kapasitas stack tak berhingga
    E. Opearsi mengosongkan stack selalu lebih cepat

8. Stack dapat berguna secara efektif (tidak redundant) untuk hal-hal berikut, **kecuali**?
    A. dalam pencarian node di dalam linked-list berdasarkan harga data
    B. dalam aplikasi sistem antrian berbasis LIFO
    C. dalam konversi algoritme rekursif menjadi nonrekursif
    D. dalam melakukan parsing nested structures
    E. dalam melakukan depth first searching

9. Dalam implementasi queue dengan circular-array, jika digunakan dua indeks front dan back, dengan front menunjukkan indeks array yang berisi **elemen berikutnya yang akan di-dequeue** dan back menunjukkan **posisi berikutnya dalam array yang akan diisi elemen yang baru masuk** saat terjadi enqueue.
   Jika diketahui front == back maka berarti:
    I. Queue kosong

II. Queue penuh
III. Queue tidak penuh ataupun kosong

A. Tidak ada yang benar
B. Semua pasti benar
C. Hanya ada satu yang benar
D. Tepat ada dua yang mungkin benar yaitu I dan II
E. Tepat ada dua yang benar yaitu II dan III

10. Kelebihan implementasi array untuk stack dibandingkan implementasi linked-list adalah kompleksitas yang lebih baik pada operasi:
    A. pop() (mengeluarkan item data top of stack).
    B. empty() (mengosongkan isi stack).
    C. isEmpty() (memeriksa apakah stack kosong).
    D. Push() (menambah item data ke dalam stack).
    E. peek() (membaca item data top of stack).

11. *Array* berisi *N* data terurut (*sorted array*) memiliki kelebihan dibandingkan *unsorted array* dalam
    A. Memungkinkan menambahkan suatu data ke dalam *array* secara *O*(1).
    B. Memungkinkan menghapus suatu data dari *array* secara *O*(*N*).
    C. Memungkinkan mengupdate suatu data secara *O*(*N*).
    D. Memungkinkan menghapus semua data secara *O*(1).
    E. Memungkinkan menemukan suatu data dalam *array* secara O(log *N*).

12. *Array* berisi *N* data tak terurut (*unsorted array*) memiliki kelebihan dibandingkan *sorted array* dalam
    A. Memungkinkan menambahkan suatu data ke dalam *array* secara *O*(1).
    B. Memungkinkan menemukan suatu data dalam *array* secara *O*(log *N*).
    C. Memungkinkan menghapus suatu data dari *array* secara *O*(*N*).
    D. Memungkinkan mengupdate suatu data secara *O*(*N*).
    E. Memungkinkan menghapus semua data secara *O*(1).

13. Masalah pada *array doubling* adalah sebagai berikut, kecuali:
    A. Setelah doubling ukuran array tetap sebesar itu walaupun ukuran data kembali mengecil (kecuali ada proses kebalikannya, eg, halving).
    B. Saat terjadi array doubling, maka diperlukan proses bersifat O(n).
    C. Saat terjadi doubling dari ukuran array semula adalah n, maka kemudian terjadi alokasi memory untuk array tersebut sebesar 3 kali n.
    D. Setiap saat memory space yang dialokasi adalah dua kali dari yang sebenarnya digunakan saat itu.
    E. Setiap penambahan data harus diperiksa apakah sudah penuh (perlu dilakukan doubling).

14. Yang bukan merupakan keuntungan sorted linked-list dibandingkan sorted array dalam menyimpan data adalah:
    A. Menghindari alokasi memori yang belum tentu digunakan.
    B. Memungkinkan operasi-operasi yang berdampak secara lokal
    C. Memungkinkan operasi pencarian logaritmis.
    D. Menghindari operasi doubling (memperbesar kapasitas dengan merealokasi memori dua kali sebelumnya).
    E. Memungkinkan penggunaan memory sesuai pemakaian sebenarnya (sebanding dengan banyaknya data).

15. Manakah pernyataan yang benar berikut ini terkait operasi penghapusan suatu data pada unsorted array (O1), sorted array (O2), unsorted linkedlist (O3), sorted linkedlist (O4), dengan O1, O2, O3, dan O4 menyatakan kompleksitas waktu algoritma pada struktur data ybs.

A. O1 <= O2 < O3 <= O4
B. O3 <= O4 < O1 <= O2
C. O4 <= O3 < O2 <= O1
D. O1 <= O2 < O3 <= O4
E. O1 = O2 = O3 = O4