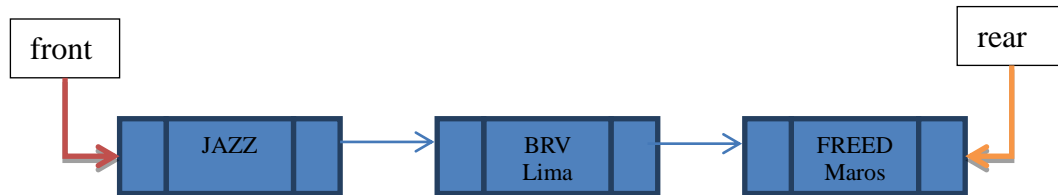
	<p align="center">UAS Semester Genap 2015/2016 DIG1G3 (IMPLEMENTASI STRUKTUR DATA) Jumat, 13 Mei 2016 Pukul 09.00 – 11.00 (120 menit) Dosen: RIM, CAH, IZM</p>			
	<p align="center">= On-Site Test, Close Book, Perorangan =</p>			
Nama Mahasiswa:	NIM:	Kelas:	Ruang:	Nilai: (Diisi Dosen)
Salinlah pernyataan berikut: <i>Saya mengerjakan ujian ini dengan jujur dan mandiri. Jika saya melakukan pelanggaran, maka saya bersedia menerima sanksi.</i>			Tanda Tangan Mahasiswa: 	
.....				

Kompetensi Dasar
Kompetensi Dasar yang akan dicapai oleh mahasiswa dengan mengikuti ujian ini: 1. KD1: Mahasiswa mampu memahami konsep singly linked list, doubly linked list, circular dan multi linked list serta stack 2. KD2: Mahasiswa mampu melakukan tracing (penelusuran) dengan baik dari potongan program yang diberikan 3. KD3: Mahasiswa mampu merancang solusi dari permasalahan yang diberikan dengan menggunakan komponen-komponen struktur data yang telah dipelajari

Peraturan Khusus Ujian
1. Anda harus menjawab SENDIRI semua pertanyaan di bawah. DILARANG KERAS untuk berkomunikasi, bekerjasama, dan meminta bantuan siapapun, melalui media apapun. 2. Gunakan lembar jawaban yang tersedia mulai dari awal s.d halaman terakhir dokumen ini. Jika lembar jawab yang disediakan masih kurang, gunakan halaman kosong di balik setiap lembar soal. Tidak ada kertas tambahan untuk menuliskan jawaban Anda. 3. Sifat ujian adalah tertutup untuk media apapun , seperti buku, catatan, dan komputer/laptop. Semua perangkat elektronik juga tidak boleh digunakan, seperti kalkulator, handphone, dll. Setiap coretan untuk perhitungan dipersilakan dilakukan di lembar soal maupun jawaban.

I. Pilihan Ganda – Hitamkan pada Lembar Jawaban yang Telah Disediakan

Soal no 1 – 3. Diberikan sebuah queue seperti gambar berikut:



1. ADT yang tepat untuk queue diatas adalah...
 - A.

```
Struct Mobil{  
    String merk;  
    Mobil *next;  
}
```
 - B.

```
Struct Merk{  
    String merk;  
    Mobil *next;  
}
```
 - C.

```
Struct Mobil{  
    String merk;  
    Merk *next;  
}
```
 - D.

```
Struct Mobil{  
    String merk;  
    String *next;  
}
```
2. Apabila dilakukan dequeue(), dequeue(), enqueue(APV), enqueue(CRV), dequeue(), maka antrian yang tepat untuk menggambarkan queue diatas adalah...
 - A. JAZZ, BRV, FREED, APV, CRV
 - B. BRV, FREED, APV, CRV
 - C. FREED, APV, CRV
 - D. APV, CRV
3. Langkah yang tepat untuk melakukan proses dequeue() pada atrian diatas adalah...
 - A.

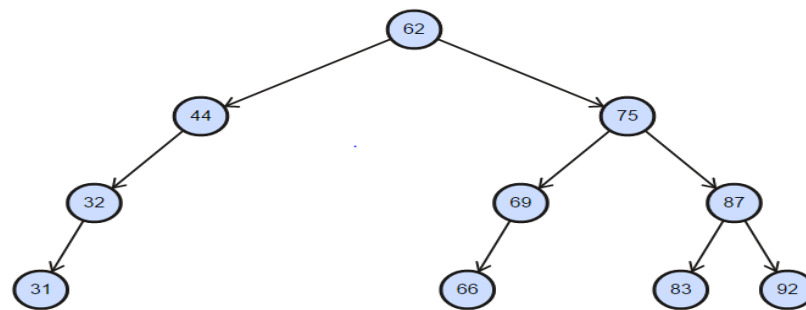
```
Mobil *temp = rear;  
rear = rear->next;  
Delete rear;
```
 - B.

```
Mobil *temp = *temp2 = front;  
While(temp->next != NULL)  
    temp = temp->next;  
rear = temp;  
temp2 = temp->next  
delete temp;
```
 - C.

```
Mobil *temp = front;  
front = front->next;  
Delete front;
```
 - D.

```
Mobil *temp = *temp2 = front;  
While(temp->next != NULL)  
    temp = temp->next;  
rear = temp;  
temp2 = temp->next  
delete temp2;
```

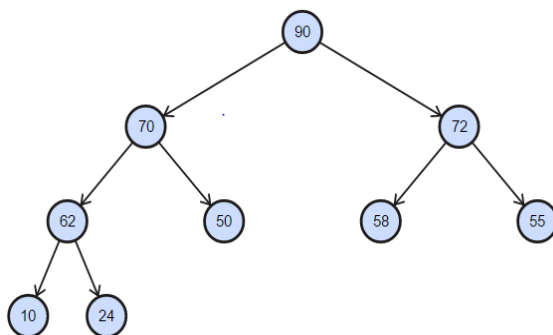
4. Hasil penelusuran pre-order untuk Biary Search Tree dibawah ini adalah...



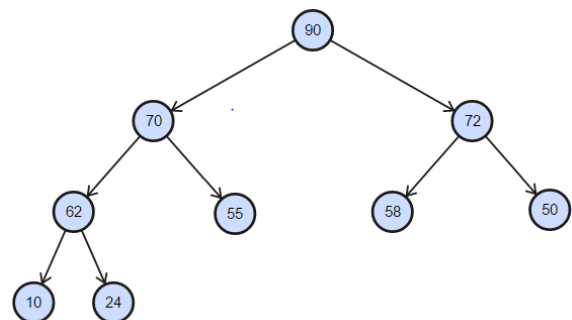
- A. 62, 44, 32, 31, 75, 69, 66, 87, 83, 92
- B. 31, 32, 44, 62, 66, 69, 75, 83, 87, 92
- C. 62, 44, 32, 31, 75, 66, 69, 87, 83, 92
- D. 31, 32, 44, 66, 69, 83, 92, 87, 75, 62

5. Gambar yang tepat untuk menggambarkan Max-Heap tree yang terbentuk dari data 90, 50, 72, 10, 62, 58, 55, 70, 24 adalah...

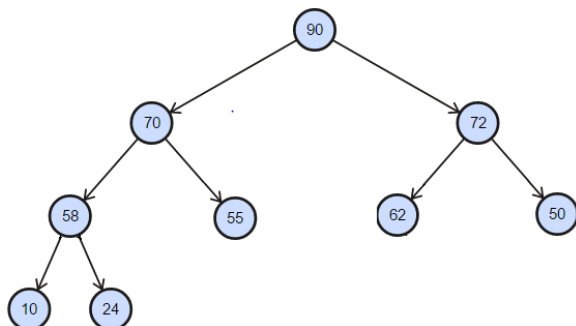
A.



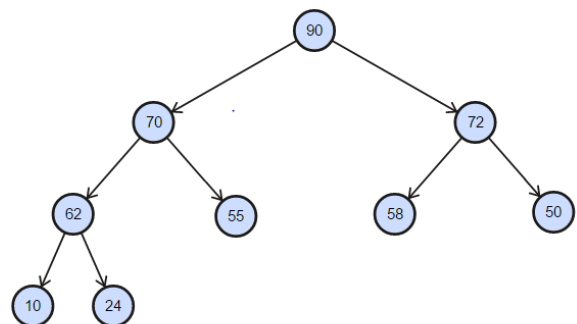
C.



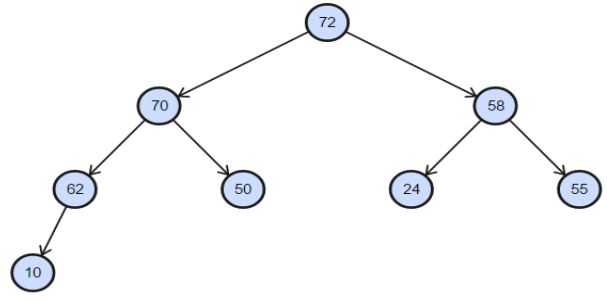
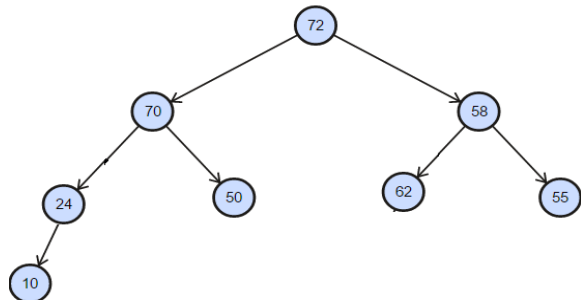
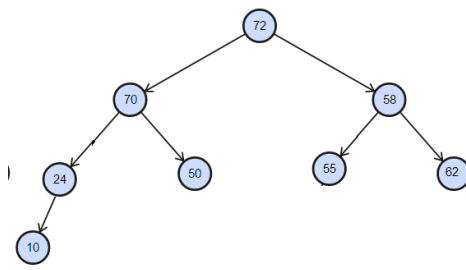
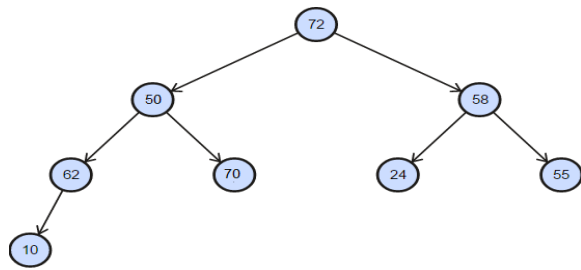
B.



D.



6. Berdasarkan data diatas, apabila dilakukan deleting terhadap node 90 pada Max-heap tree, maka tree yang terbentuk adalah...

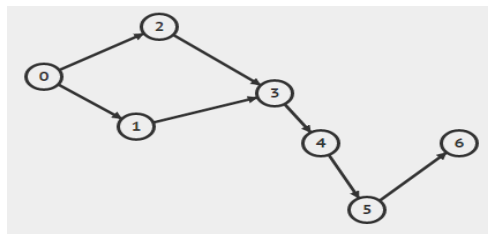


Soal untuk no 6, 7, 8, dan 9

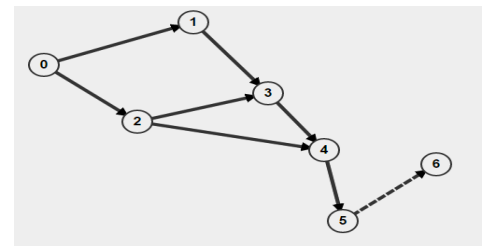
Vertex	1	2	3	4	5	6
1	0	1	1	0	0	0
2	0	0	0	1	0	0
3	0	0	0	1	0	0
4	0	0	0	0	1	1
5	0	0	0	0	0	1
6	0	0	0	0	0	0

7. Berdasarkan matrix diatas, maka graph yang dibentuk adalah?

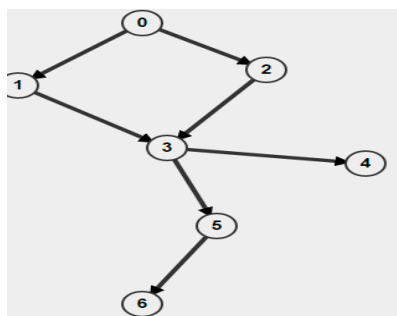
A.



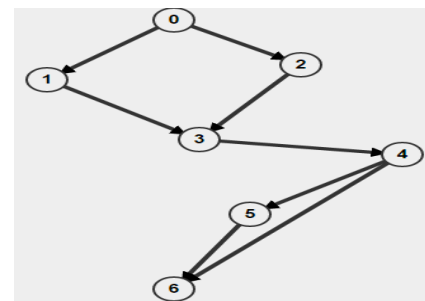
C.



B.

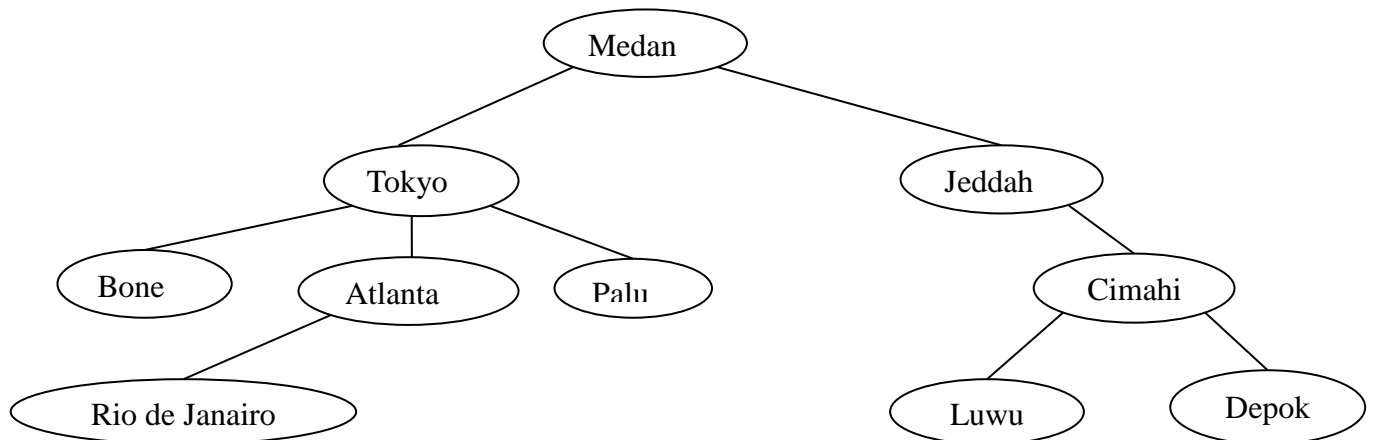


D.



8. Hasil BFS untuk graph diatas adalah...
 - A. 0, 1, 2, 3, 4, 5, 6
 - B. 0, 1, 3, 4, 5, 6, 2
 - C. 6, 5, 4, 3, 1, 2, 0
 - D. 6, 4, 5, 3, 2, 1, 0
9. Hasil DFS untuk graph diatas adalah...
 - A. 0, 1, 2, 3, 4, 5, 6
 - B. 0, 1, 3, 4, 5, 6, 2
 - C. 6, 5, 4, 3, 1, 2, 0
 - D. 6, 4, 5, 3, 2, 1, 0
10. Topological sort dari graph diatas adalah....
 - A. 0, 1, 2, 3, 4, 5, 6
 - B. 0, 1, 3, 4, 5, 6, 2
 - C. 6, 5, 4, 3, 1, 2, 0
 - D. 6, 4, 5, 3, 2, 1, 0

Pohon berikut digunakan untuk soal No.11 – 14



11. *Predecessor* bagi *node* dengan *key* Luwu adalah
 - A. Depok
 - B. Cimahi
 - C. Medan, Jeddah, Cimahi
 - D. Medan, Tokyo, Atlanta
12. Internal *node* dari pohon tersebut adalah
 - A. Rio de Janeiro, Luwu, Depok
 - B. Bone, Rio de Janeiro, Palu, Luwu, Depok
 - C. Tokyo, Atlanta, Jeddah, Cimahi
 - D. Medan, Tokyo, Jeddah, Atlanta, Cimahi
13. Level dari *node* dengan *key* Cimahi adalah...
 - A. 2
 - B. 1

- C. 0
- D. 3

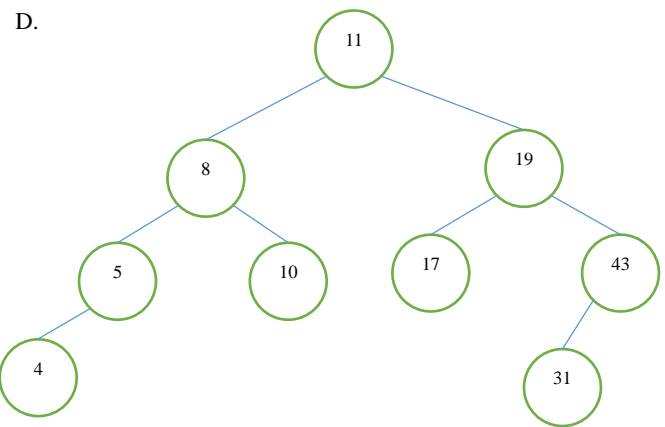
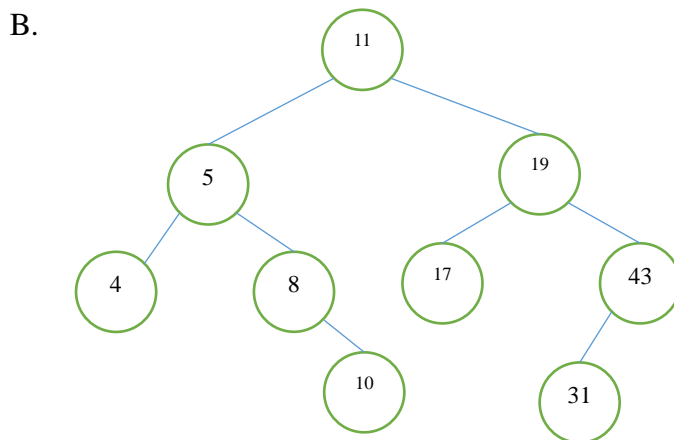
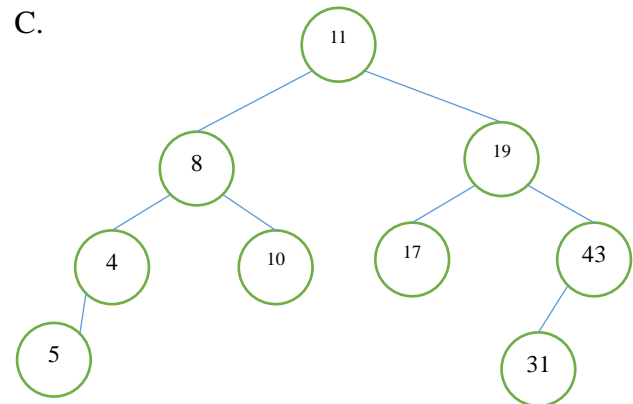
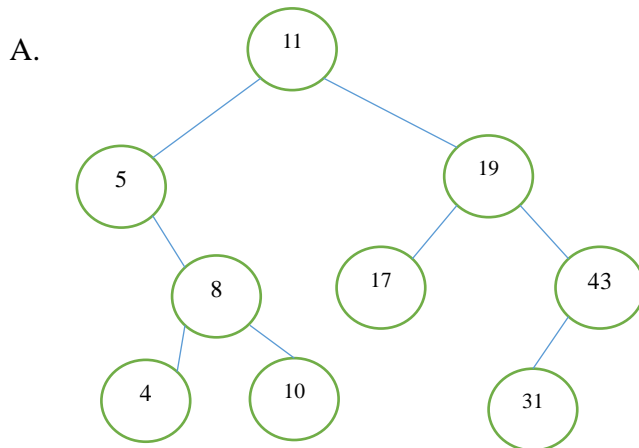
14. Hasil penelusuran inorder dari pohon tersebut adalah...

- A. Medan – Tokyo – Bone – Atlanta – Rio de Janeiro – Palu – Jeddah – Cimahi – Luwu – Depok
- B. Bone – Tokyo – Rio de Janeiro – Atlanta – Palu – Medan – Jeddah – Luwu – Cimahi – Depok
- C. Bone – Rio de Janeiro – Atlanta – Palu – Tokyo – Luwu – Depok – Cimahi – Jeddah – Medan
- D. Medan – Tokyo – Jeddah – Bone – Atlanta – Palu – Cimahi – Rio de Janeiro – Luwu – Depok

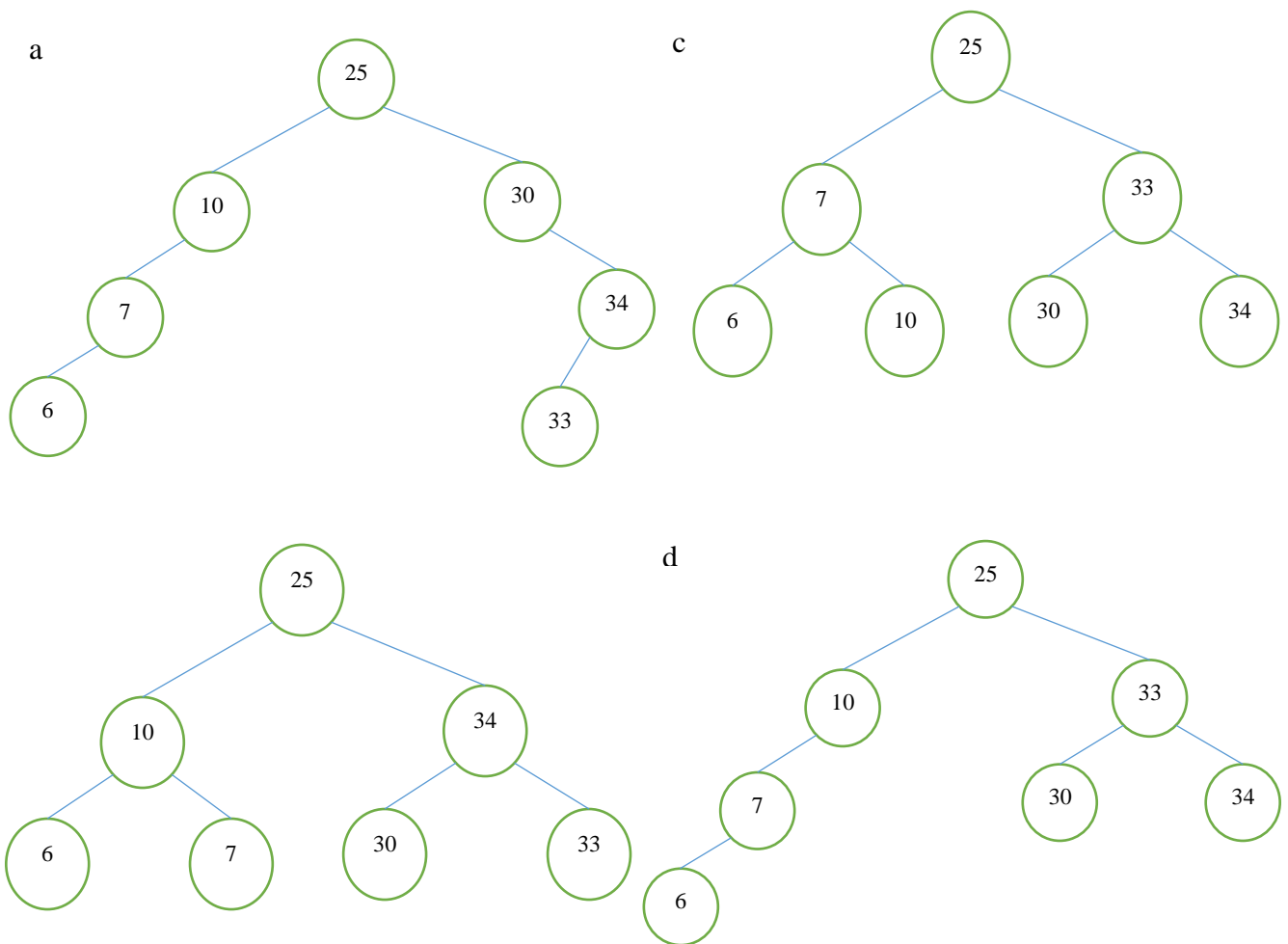
15. Heaptree dari tree berdasarkan array 1, 7, 16, 5, 18, 31, 32, 9, 16, 46 adalah

- A. 46, 32, 31, 18, 16, 16, 9, 7, 5, 1
- B. 46, 18, 32, 16, 7, 31, 16, 9, 5, 1
- C. 46, 32, 31, 18, 16, 9, 7, 5, 1
- D. 46, 18, 31, 16, 7, 32, 16, 9, 5, 1

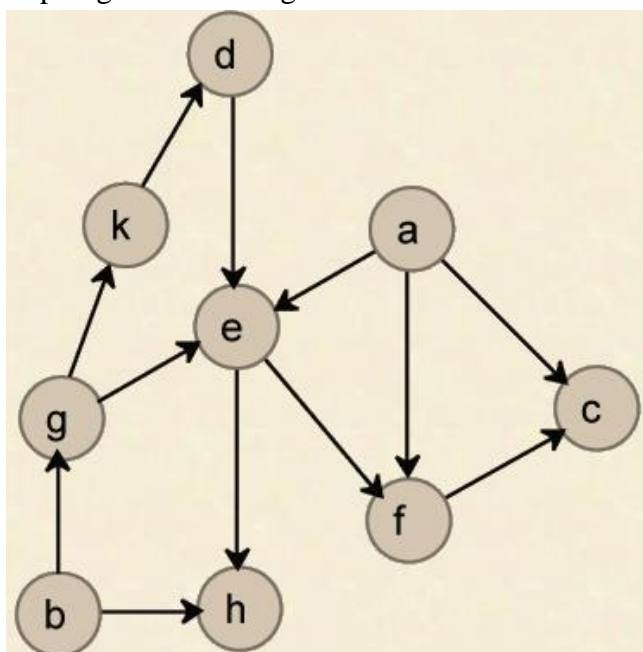
16. Jika pada BST dengan masukan 11, 6, 19, 4, 8, 17, 43, 5, 10, 31 dilakukan penghapusan pada node dengan key 6, bentuk BST akan menjadi...



17. Jika sebuah pohon AVL memiliki masukan berikut: 25, 10, 30, 34, 33, 7, 6 hasil akhir dari pohon tersebut adalah

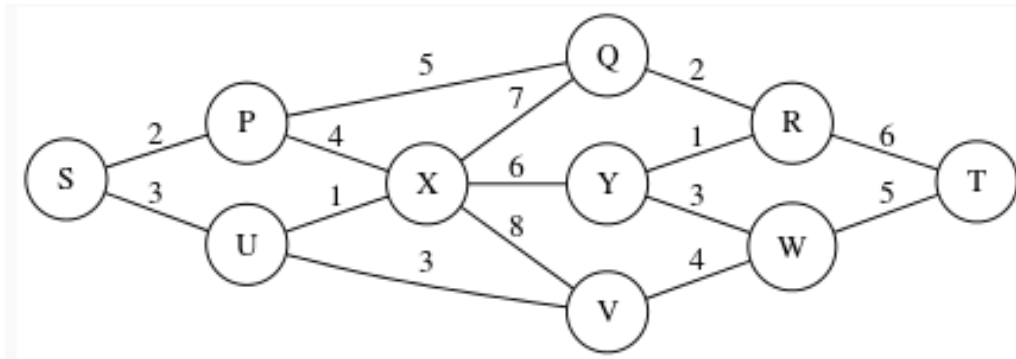


18. Topological sort dari graf berikut adalah:



- A. b g k d a e h f c
- B. a b g k d c f h e
- C. b g k d c e f h a
- D. b g a k d e h f c

19. Hasil penelusuran BSF dari graf berikut jika dimulai dari node dengan key X adalah



- A. X, P, U, Q, R, Y, V, S, W, T
- B. X, Y, Q, V, P, U, T, S, R, W
- C. X, Q, Y, V, P, U, R, W, S, T
- D. X, Q, Y, V, U, P, S, R, W, T

20. Berikut merupakan pernyataan yang tepat mengenai queue:

- A. Queue bersifat LIFO, dengan enqueue merupakan primitif untuk memasukkan data dan dequeue merupakan primitif untuk menghapus data
- B. Queue bersifat FIFO, dengan enqueue merupakan primitif untuk memasukkan data dan dequeue merupakan primitif untuk menghapus data serta data dimasukkan dari sisi depan (front) array
- C. Queue bersifat FIFO, dengan enqueue merupakan primitif untuk memasukkan data dan dequeue merupakan primitif untuk menghapus data serta data dimasukkan dari sisi belakang (rear) array
- D. Queue bersifat FIFO, dengan enqueue merupakan primitif untuk memasukkan data dan dequeue merupakan primitif untuk menghapus data serta data dihapus dari sisi belakang (rear) array

II. Essay

1. Perhatikan definisi queue berikut ini!

```
struct Node {
    char NIM[13];
    char Nama[30];
    Node *Next;
};

struct Queue {
    Node *Front;
    Node *Rear;
};
```

Queue di atas diinisialisasi menggunakan prosedur inisialisasi sebagai berikut!

```
void inisialisasi(Queue *Q) {
    Q->Front = NULL;
    Q->Rear = NULL;
}
```

}

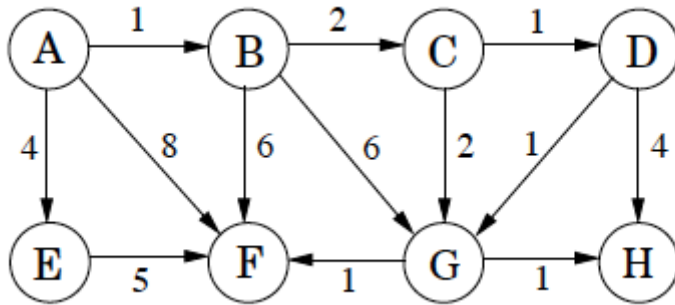
Tuliskan implementasi dari prosedur enqueue dan dequeue dari queue di atas.

```
void enqueue(Queue *Q, char NIM[13], char Nama[30]);
```

```
void dequeue(Queue *Q);
```

2. Buatlah AVL tree berdasarkan angka berikut secara berturut-turut: 8, 4, 9, 11, 10, 3, 2, 5, 6, 4. Tuliskan dalam langkah per langkah! Setelah AVL tree tersebut jadi, traversal tree tersebut secara pre-order, in-order dan post-order.

Untuk soal 3 – 5, perhatikan graph berikut!



Sumber: <http://www.chegg.com/homework-help/questions-and-answers/suppose-dijkstra-s-algorithm-run-following-graph-starting-node--b-show-nal-shortest-path-t-q3684344>

3. Tuliskan hasil traversal graph di atas secara DFS dan BFS, dimulai dari node A.
4. Buatlah topological sort dari graph di atas.
5. Hitunglah rute terpendek graph di atas menggunakan algoritma Dijkstra, dimulai dari node A.