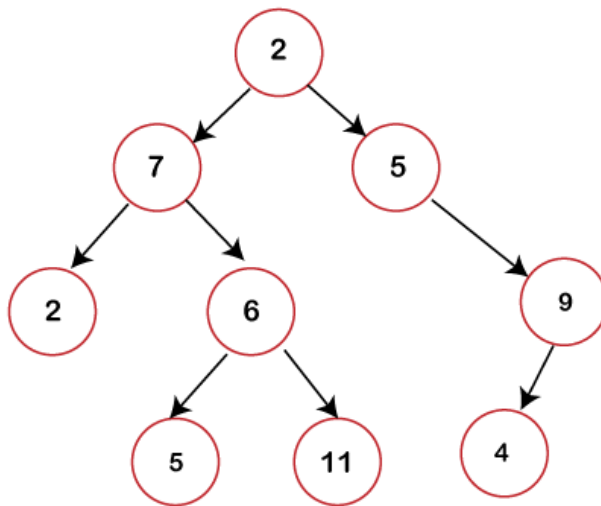


Soal Latihan Pra-Assessment 02 ISD

Berdasarkan gambar tree berikut (sumber: <https://www.javatpoint.com/binary-tree-vs-binary-search-tree>) :



Buatlah:

1. Penelusuran berdasarkan pre-order.
2. Penelusuran berdasarkan in-order.
3. Penelusuran berdasarkan post-order.
4. Sebutkan depth dari tree, level dari node 9, sibling dari node 6, ancestor node 11 dan child dari node 5.
5. Buatlah suatu heap-tree (prioritas maksimum) berdasarkan masukan data berikut:

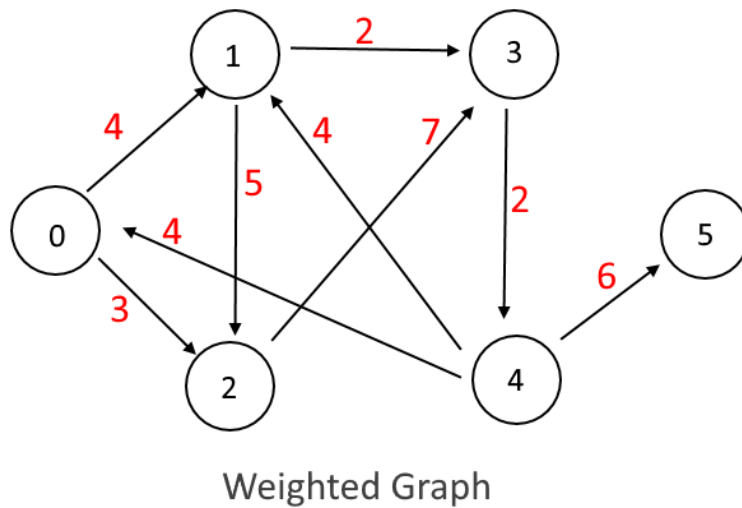
3	23	12	7	45	8	34	17	35	55
---	----	----	---	----	---	----	----	----	----

6. Jika pada heap tree berdasarkan no 5 ditambahkan data 60, bagaimanakah tree yang terbentuk?
7. Buatlah heapsort berdasarkan tree yang terbentuk pada no 6. Lakukan per-langkah!
8. Buatlah suatu binary search tree (BST) berdasarkan masukan data berikut:

19	23	11	7	45	8	34	3	12	55
----	----	----	---	----	---	----	---	----	----

9. Berdasarkan BST yang terbentuk pada soal no 8, lakukan penghapusan untuk node berikut (lakukan berturut-turut): 8, 7, 19

Berdasarkan gambar graf berikut (sumber: <https://algorithms.tutorialhorizon.com/weighted-graph-implementation-java/>) :



Buatlah:

10. Representasi graf berdasarkan matrix adjacency
11. Representasi graf berdasarkan list adjacency
12. Representasi graf berdasarkan matrix incidencey

Noted:

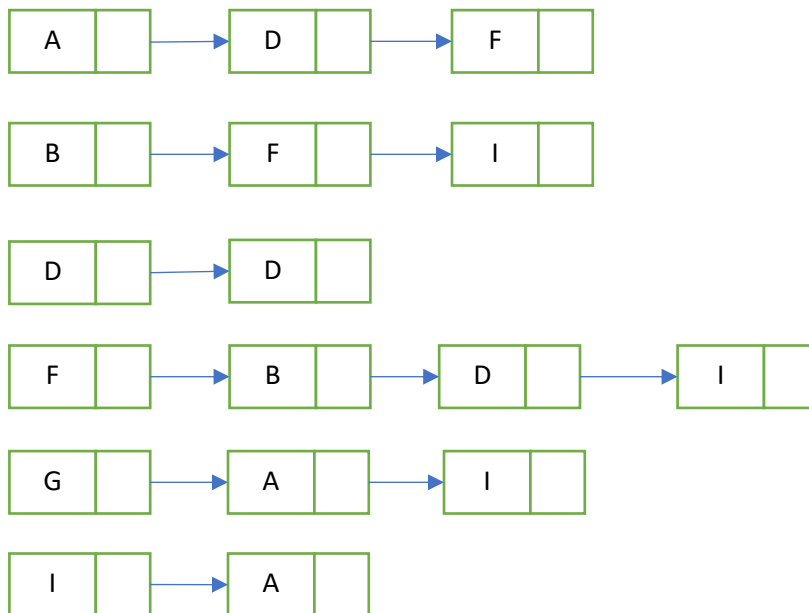
Daftar edge untuk matriks di atas:

Vertex awal	Vertex akhir	Edge
0	1	e1
0	3	e2
1	2	e3
1	3	e4
2	3	e5
3	4	e6
4	0	e7
4	1	e8
4	5	e9

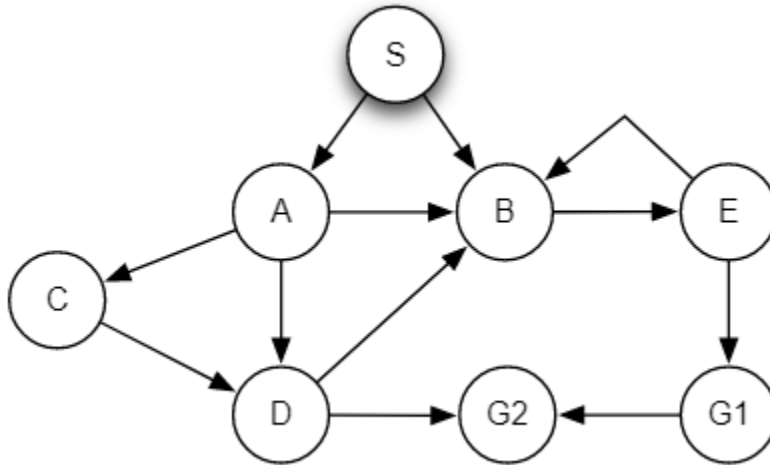
13. Gambarkan graf yang terbentuk berdasarkan matrix adjacency berikut

	1	2	3	4	5	6	7
1	0	1	0	0	1	1	0
2	1	0	1	0	1	0	0
3	0	1	0	0	1	1	1
4	0	0	0	0	0	0	1
5	1	1	1	0	0	1	1
6	1	0	1	0	1	0	0
7	0	0	1	1	1	0	0

14. Gambarkan graf yang terbentuk berdasarkan list adjacency berikut



15. Berdasarkan graf berikut, tuliskan penelusuran BFS dan DSF dengan vertex source pada node S  
(sumber: <https://people.cs.umass.edu/~liberato/courses/2016-fall-compsci190d/homeworks/19-directed-graph-search-again/>) !



Diketahui pasangan *key* dan *value* sebagai berikut:

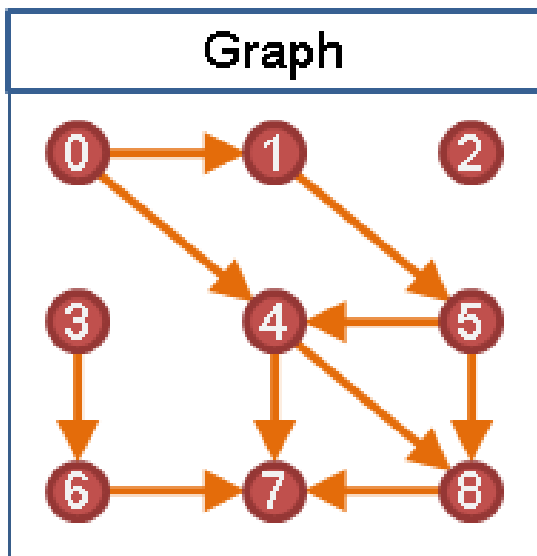
KEY	VALUE
27	Ketupat
113	Opor
7	Sayur Godog
18	Sayur Lodeh
4	Rendang
200	Kentang Mustofa
96	Sambel Goreng Udang

Gambarkan isi *hash table* untuk pasangan *key-value* tersebut ke dalam suatu table yang awalnya masih kosong. Ingat, karena *value* diketahui, yang akan mengisi *table* adalah *value*-nya, sesuai dengan *hash code* dari *key*.

Misal, key 27 memiliki *hash code* 3, jadi pada table indeks ke-3 akan terisi “Ketupat” (bukan 27). Tuliskan terlebih dahulu *hash code* yang terbentuk untuk setiap *key* pada masing-masing soal. Gunakan ketentuan berikut untuk penanganan *collision* dan *hash function*. (Silahkan gunakan kalkulator jika dibutuhkan)

**Implementasi Struktur Data**  
**Genap 2024/2025**

16. *Value* akan dipetakan pada *table* dengan ukuran 5. *Hash function* bagi *key* adalah  $h(k) = (11 * k) \bmod M$ .  $M$  merupakan besar *table* dengan  $k$  adalah nilai *key*. Penanganan *collision* menggunakan *separate chaining*.
17. *Value* akan dipetakan pada *table* dengan ukuran 7. *Hash function* bagi *key* adalah  $h(k) = k \bmod M$ .  $M$  merupakan besar *table* dengan  $k$  adalah nilai *key*. Penanganan *collision* menggunakan *linear probing*.
18. *Value* akan dipetakan pada *table* dengan ukuran 11. *Hash function* bagi *key* adalah  $h(k) = k \bmod M$ .  $M$  merupakan besar *table* dengan  $k$  adalah nilai *key*. Penanganan *collision* menggunakan *quadratic probing*.
19. *Value* akan dipetakan pada *table* dengan ukuran 11. *Hash function* bagi *key* adalah  $h_1(k) = (11 * k) \bmod M$  dan  $h_2(k) = (k \bmod 3) + 1$ .  $M$  merupakan besar *table* dengan  $k$  adalah nilai *key*. Penanganan *collision* menggunakan *double hashing*.
20. Buatlah topological sort dari graf berikut. Jelaskan langkah pembuatannya!



Sumber gambar: <https://web.ntnu.edu.tw/~algo/DirectedAcyclicGraph.html>