UNIVERSITAS INDONESIA
FAKULTAS ILMU KOMPUTER
Veritas, Probitas, Justitia

Pengajar: Suryana Setiawan, Dipta Tanaya,
Iis Solichah, Pudy Prima

CSGE602040–Struktur Data dan Algoritma
UAS Semester Gasal 2022/2023
15 Desember 2022, 13:00 - 15:30

PERNYATAAN KESANGGUPAN MENTAATI TATA TERTIB UJIAN

"Saya telah membaca dan memahami ketentuan tata tertib berikut ini, serta menyatakan bahwa jawaban ujian ini adalah hasil pekerjaan saya sendiri. Saya menyetujui jika melakukan pelanggaran atas ketentuan tersebut, saya bersedia diproses sesuai ketentuan yang berlaku (SK Dekan 103a Tahun 2020) dengan sanksi maksimal **nilai akhir E**."

Nama & Tanda-tangan: Kelas: Nomor Pokok Mahasiswa:

TATA TERTIB UJIAN

- Semua alat komunikasi elektronik dalam kondisi non-aktif (dimatikan), dimasukkan ke dalam tas dan diletakkan pada tempat yang telah disediakan.
- Peralatan ujian yang boleh dibawa adalah alat tulis dan yang diperbolehkan sesuai sifat ujian.
- Peserta ujian menempati tempat duduk yang telah ditentukan.
- Peserta ujian menuliskan nama dan NPM pada setiap lembar jawaban ujian.
- Peserta mulai membuka soal dan mengerjakan ketika pengawas mengatakan ujian dimulai dan berhenti bekerja (meletakkan alat tulis) ketika pengawas mengatakan waktu habis.
- Peserta tidak berkomunikasi dalam bentuk apapun dengan peserta lain selama berada di ruang ujian, termasuk pinjam meminjam alat tulis, serta tidak memberi atau menerima bantuan dari siapapun selama ujian berlangsung.
- Peserta yang meninggalkan ruang ujian dianggap selesai mengerjakan. Jika karena kondisi medis khusus tidak bisa memenuhi ketentuan ini, peserta wajib melaporkan kepada pengawas sebelum ujian dimulai.
- Setelah selesai mengerjakan atau setelah waktu habis, peserta segera meninggalkan berkas soal dan lembar jawaban ujian di meja masing-masing, mengambil tas dan segera keluar tanpa mengganggu peserta lain serta tanpa berkomunikasi dengan peserta lain.
- Jawaban ujian ini tidak akan dinilai jika pernyataan di atas ini tidak ditandatangani.

Informasi Tambahan

- Diperbolehkan membawa **note ditulis tangan** pada maksimal 1 lembar kertas berukuran A4.
- Tuliskan jawaban menggunakan **pulpen hitam/biru**.
- Beri **nama**, **NPM**, dan **kelas** di **semua lembar berkas ujian**.
- Soal boleh dicoret-coret.

Nama: _____ NPM: _____ Kelas: _____

Lembar Jawab untuk Soal Bagian A (25 soal)

Berikan tanda silang (X) pada opsi jawaban yang paling tepat. Gunakan tip-ex untuk mengganti jawaban.
Jawaban Bagian A yang ditulis di luar lembar jawab ini tidak akan dinilai.

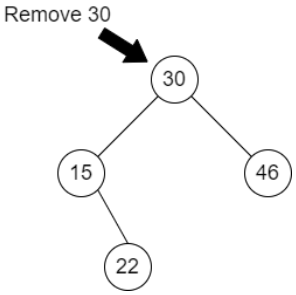
1.	A B C D	11.	A B C D
2.	A B C D	12.	A B C D
3.	A B C D	13.	A B C D
4.	A B C D	14.	A B C D
5.	A B C D	15.	A B C D
6.	A B C D	16.	A B C D
7.	A B C D	17.	A B C D
8.	A B C D	18.	A B C D
9.	A B C D	19.	A B C D
10.	A B C D	20.	A B C D
		21.	A B C D
		22.	A B C D
		23.	A B C D
		24.	A B C D
		25.	A B C D

Bagian A. Pilihlah Jawaban yang Tepat

1. Manakah pernyataan yang **benar** mengenai AVL tree?
   a. Dalam AVL tree, perbedaan tinggi subpohon kiri dan subpohon kanan paling besar adalah dua.
   b. Worst case time complexity untuk operasi penghapusan elemen pada AVL tree adalah O(N), dengan N merepresentasikan banyaknya data pada tree.
   c. Penelusuran AVL tree secara level-order akan menghasilkan data yang terurut menaik.
   d. Penambahan elemen baru pada AVL tree mungkin menyebabkan perubahan elemen root.

2. Berapa **tinggi maksimum** AVL tree yang terdiri dari **12 node**? (Catatan: AVL tree yang terdiri dari 1 node memiliki tinggi 0.)
   a. 3
   b. 4
   c. 5
   d. 6

3. Perhatikan AVL tree berikut. Jika dilakukan **penghapusan elemen 30** dengan kaidah **successor in-order**, maka yang terjadi adalah …



   a. Perlu dilakukan single rotation
   b. Perlu dilakukan double rotation
   c. Perlu dilakukan baik single maupun double rotation
   d. Tidak perlu melakukan proses rotasi

4. Pada sebuah AVL tree kosong akan dilakukan proses penambahan elemen-elemen berikut secara berurutan: 39, 26, 15, 18, 27, 30. Berapa banyak **rebalancing** yang terjadi?
   a. Terjadi 2 kali rebalancing, keduanya single rotation
   b. Terjadi 2 kali rebalancing, keduanya double rotation
   c. Terjadi 2 kali rebalancing, 1 single rotation dan 1 double rotation
   d. Terjadi 3 kali rebalancing, 2 single rotation dan 1 double rotation

Soal no 5 dan 6 mengacu pada penjelasan berikut ini.

Diketahui sebuah array yang merepresentasikan **binary heap** dengan kaidah **min-heap.** Struktur ini sudah berisikan sejumlah data bilangan unik, dan dalam tabel ini bilangan-bilangan itu masing-masing diwakili oleh sebuah variabel huruf (perhatikan bahwa urutan huruf bukan urutan besar-kecilnya data).

A	B	C	D	E	F	G	H	I

5. Manakah pernyataan yang pasti benar?
   a. Nilai C selalu lebih kecil dibandingkan H maupun I.
   b. Node F tidak memiliki anak.
   c. Leaf pertama dalam representasi array adalah H.
   d. Nilai H atau I adalah yang paling besar.

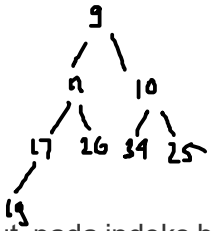
6. Manakah pernyataan yang pasti benar?
   a. Jika B > C dan F < G, dan terjadi getMin(), setelah percolate down I berhenti di posisi yang sebelumnya ditempati oleh G.
   b. Jika B > C dan F > G, dan terjadi getMin(), setelah percolate down I berhenti di posisi yang sebelumnya ditempati oleh F.
   c. Jika B < C dan D > E, dan terjadi getMin(), setelah percolate down I berhenti di posisi yang sebelumnya ditempati oleh E.
   d. Jika B < C dan D < E, dan terjadi getMin(), setelah percolate down I berhenti di posisi yang sebelumnya ditempati oleh H.

7. Suatu priority queue dengan kaidah **minheap**, yang sudah berisikan **N** bilangan bulat. Untuk mencari bilangan **terbesar ke k** di mana 1 <= k <= N, operasi priority queue akan digunakan (operasi beberapa kali getMin(), bukan dengan algoritma pencarian lain!) maka menjadi berapakah kompleksitas untuk **kasus terburuk**?
   a. O(N log N) dengan k = N
   b. O(log N) dengan k = N
   c. O(log N) dengan k = 1
   d. O(N log N) dengan k = 1

   ga baca terbesar jir

8. Diketahui sebuah array yang merepresentasikan sebuah **binary heap** dengan kaidah **min-heap** sebagai berikut (indeks array dari 0, 1, ...)

indeks	0	1	2	3	4	5	6	7
data	10	17	25	19	26	34		

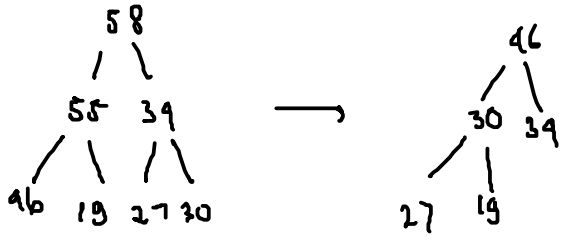


   Jika dilakukan penambahan (*insert*) dua elemen yaitu 9 dan 11 berturut-turut, pada indeks berapakah kemudian 11 ditempatkan?
   a. 0
   b. 1
   c. 2
   d. 3

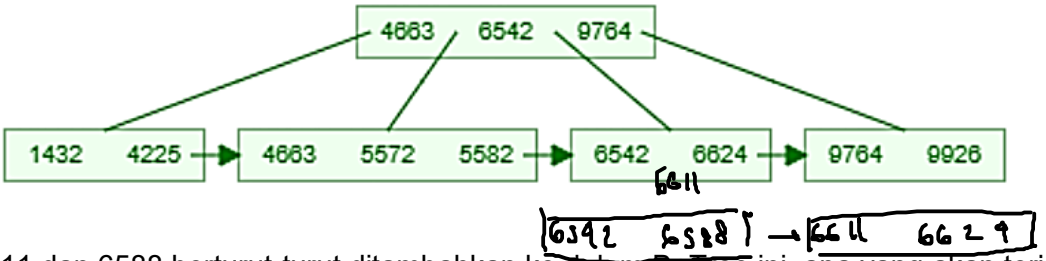
9. **Heapsort** digunakan untuk melakukan sorting secara **ascending** data berikut:

indeks	0	1	2	3	4	5	6
data	19	58	27	46	55	34	30

   Setelah dilakukan heapify (fix-heap) kemudian iterasi memindahkan max ke sorted area sebanyak dua kali, berapakah data yang berada pada **indeks 4**?
   a. 34
   b. 30
   c. 27
   d. 19



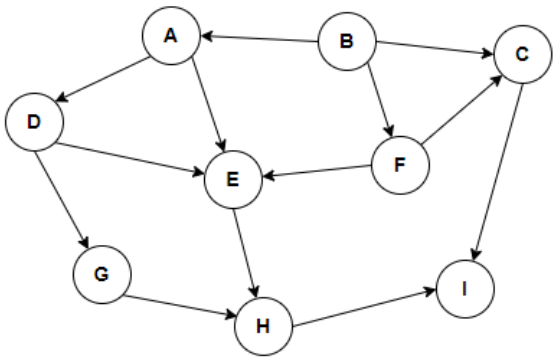
10. Diberikan B+Tree dengan max-degree = 4 berikut:



Data 6611 dan 6588 berturut-turut ditambahkan ke dalam B+Tree ini, apa yang akan terjadi (setelah keduanya ditambahkan)?

a. Terbentuknya root baru yang berisi 6588.
b. Terbentuknya root baru yang berisi 6611.
c. Tidak terbentuk root baru maupun internal node baru karena leaf node masih dapat menampung kedua data tersebut tanpa splitting.
d. Splitting hanya terjadi di leaf saja.

Graph berikut untuk nomor 11 - 13



B → {A, F} → C, D, E → I, G, H

11. Perhatikan graf pada gambar. Lakukan traversal dengan DFS dimulai dari verteks A (artinya, A adalah verteks ke-1 yang di-visit). Urutan pengecekan verteks yang adjacent adalah secara reverse-alphabetical (dari Z ke A). Verteks manakah yang ke-6 di-visit?

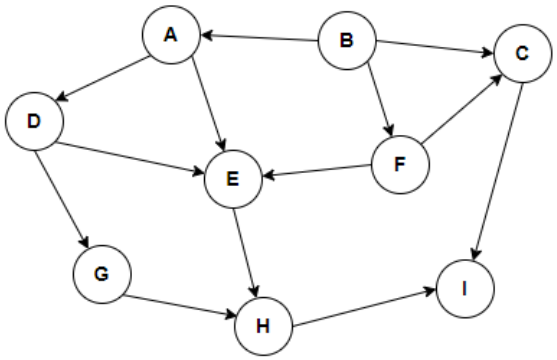
a. Verteks D
b. Verteks E
c. Verteks G
d. Verteks I

12. Perhatikan graf pada gambar. Sebuah traversal dengan BFS dimulai dari verteks B (artinya, B adalah verteks ke-1 yang di-visit), lalu verteks yang ke-5 di-visit adalah D, dan verteks yang ke-7 di-visit adalah I. Urutan pengecekan verteks yang adjacent adalah secara random. Verteks manakah yang terakhir di-visit?

a. Verteks D
b. Verteks G
c. Verteks H
d. Verteks I

13. Perhatikan graf pada gambar. Urutan verteks mana yang merupakan sebuah traversal dengan BFS?
    a. ADEGHI ✓
    b. DGHIE
    c. EHFIC
    d. FCEIB

A D E G H



14. Struktur data Queue cocok digunakan sebagai struktur data utama untuk mengimplementasikan algoritma …
    a. DFS dengan urutan pengecekan verteks yang adjacent adalah dari A-Z
    b. BFS dengan urutan pengecekan verteks yang adjacent adalah secara random
    c. Prim's minimum spanning tree dengan semua edge berbobot unik
    d. *Cycle detection* pada graf berarah tak berbobot

15. Data ruas jalan antar kota di Provinsi P (selanjutnya disebut: ruas jalan) direpresentasikan dengan model graph. Verteks merepresentasikan kota, dan edge merepresentasikan ruas jalan yang menghubungkan (bolak-balik) kota X dan kota Y. Dengan demikian disebut juga bahwa kota X dan kota Y saling bertetangga. Graf tersebut diimplementasikan dengan Adjacency List dengan urutan kota bersifat alfabetis (baik horizontal maupun vertikal pada tabel, lihat ilustrasi). Masing-masing kota memiliki daftar tetangga kota namun tidak menyimpan jumlah tetangga.

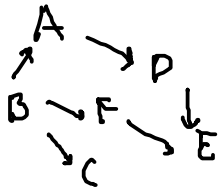
Kota	**Tetangga Kota**
Banyuwangi	*Bondowoso, Jember, Situbondo*
Blitar	*Kediri, Malang, Tulungagung*
Bojonegoro	*Madiun, Nganjuk, Ngawi, Lamongan, Tuban*
Bondowoso	*Banyuwangi, Jember, Situbondo*
… dst.	*… dst.*

Jika banyaknya kota adalah N, dan banyaknya ruas jalan adalah M, berapakah kompleksitas (dalam Big Oh) dari query:
"Kota manakah yang jumlah tetangganya terbanyak?"

    a. O (N)
    b. O (M)
    c. O (NM)
    d. O (N+M)

16. Diberikan sebuah graf berarah dalam representasi **Edge List** sebagai berikut:

E = {(A, D), (B, A), (B, C), (B, F), (D, E), (F, E))}



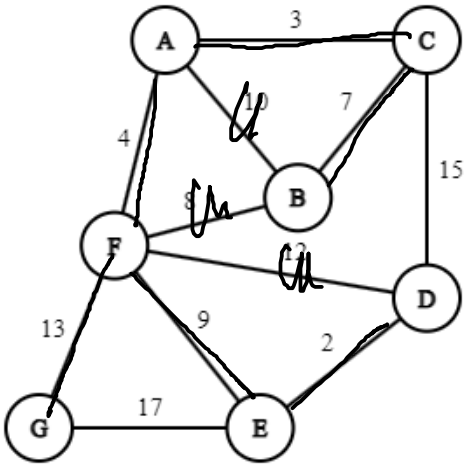
Manakah pernyataan berikut ini yang **BENAR**?

a. BFS mulai dari verteks A dapat mem-visit semua verteks pada graf
b. BFS mulai dari verteks B dapat mem-visit semua verteks pada graf
c. DFS mulai dari verteks manapun tidak dapat mem-visit verteks C
d. DFS mulai dari verteks manapun tidak dapat mem-visit verteks D

17. Pernyataan yang **tidak tepat** mengenai algoritma Dijkstra untuk connected graph adalah

a. Semua edge yang ada pada graf harus memiliki bobot non-negatif
b. Path yang memiliki jarak terpendek selalu melewati jumlah verteks yang paling sedikit
c. Algoritma Dijkstra dapat menghitung jarak terpendek ke semua verteks dalam graf dari sebuah verteks sumber
d. Verteks yang dipilih untuk masuk ke area hijau adalah verteks di area abu-abu dengan jarak sementara yang terpendek

Graph berikut untuk nomor 18 - 21



9 + 3 + 7 + 9 + 2 + 13

18. Algoritma Dijkstra diterapkan untuk mencari jarak terpendek dari verteks C ke semua vertex yang ada di dalam graph. Berikut ini pernyataan yang **tepat** mengenai hal tersebut adalah

a. Jarak terpendek verteks C ke E adalah 17
b. Verteks pertama yang diketahui jarak terpendeknya setelah C adalah B
c. Verteks yang diketahui jarak terpendeknya setelah F dan B adalah D
d. Verteks E merupakan verteks yang terakhir diketahui jarak terpendeknya dari C

19. Algoritma Prim's diterapkan untuk mencari Minimum Spanning Tree pada graf di atas. Algoritma dimulai dari verteks A. Berikut ini pernyataan yang **salah** mengenai hal tersebut adalah …

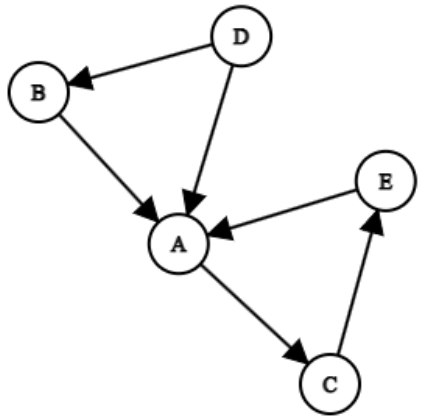
a. Total bobot MST yang terbentuk adalah 38
b. Edge pertama yang ditambahkan ke MST adalah AC dengan bobot 3
c. Verteks yang predecessornya verteks A adalah B, C, dan F
d. Verteks D terhubung dengan MST melalui edge DE dengan bobot 2

20. Algoritma Kruskal diterapkan untuk mencari Minimum Spanning Tree pada graf di atas. Edge pertama yang di-reject karena membentuk cycle adalah …

a. AB
b. BC
c. DF
d. BF

21. Algoritma Kruskal diterapkan untuk mencari Minimum Spanning Tree pada graf di atas. Edge yang tidak diperiksa karena edge pada MST sudah mencapai |V| - 1, yaitu 6 edges, adalah …
    a. GF, DC, dan EG
    b. EG dan DC
    c. BF, AB, dan DF
    d. DF, FG, DC, dan EG

22. Perhatikan graf berarah di bawah ini.



Berikut ini pernyataan yang benar mengenai Topological Sorting dari graf tersebut adalah.
    a. Topological sorting graf tersebut tidak dapat ditentukan
    b. Topological sorting yang dapat terbentuk adalah DBEAC
    c. Jika edge AC dihapus, maka topological sorting yang terbentuk adalah DBCAE
    d. Jika edge EA dihapus, topological sorting dapat dimulai dari verteks D atau C

23. Manakah pernyataan yang **benar** mengenai struktur data hash table?
    a. Hash function digunakan untuk menentukan indeks penempatan suatu data pada hash table.
    b. Double hashing berpotensi mengakibatkan terjadinya secondary clustering pada hash table.
    c. Jika suatu hash table berukuran m dan terisi sebanyak n data, maka nilai load factornya adalah m/n.
    d. Collision adalah kondisi saat dua data yang sama dipetakan ke indeks yang berbeda pada hash table.

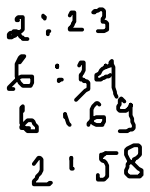
24. Diberikan sebuah hash table kosong berukuran 11. Hash table tersebut menggunakan pendekatan collision resolution dengan double hashing. Fungsi hash yang digunakan adalah sebagai berikut.
    $h_1(x) = x \bmod 11$
    $h_2(x) = (x \bmod 7) + 3$

Elemen-elemen berikut akan ditambahkan ke dalam hash table tersebut secara berurutan: 23, 45, 12, 40.
Di indeks manakah data 40 akan disimpan?

a. 1
b. 4
c. 7
d. 9

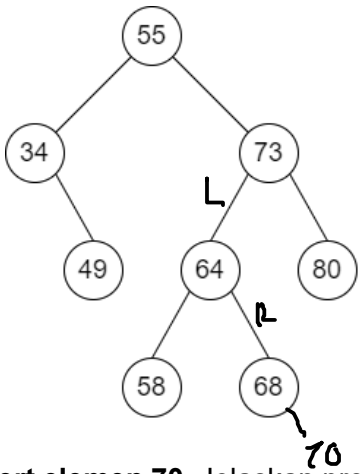
25. Diketahui suatu hash table menggunakan pendekatan collision resolution dengan **open hashing (separate chaining)**. Struktur data chaining yang digunakan adalah **linked list**, di mana setiap penambahan data baru akan disisipkan sebelum elemen pertama (**insert first**). Ukuran array hash table tersebut adalah 7, dan fungsi hash yang digunakan adalah *h(x) = x mod 7*, dengan x merupakan nilai data yang disimpan.
Berangkat dari hash table yang masih kosong, akan dilakukan penambahan data berikut secara berurutan: 23, 34, 69, 58, 44.
Elemen manakah yang akan menjadi elemen **next** dari **58**?

a. 23
b. 34
c. 69
d. 44

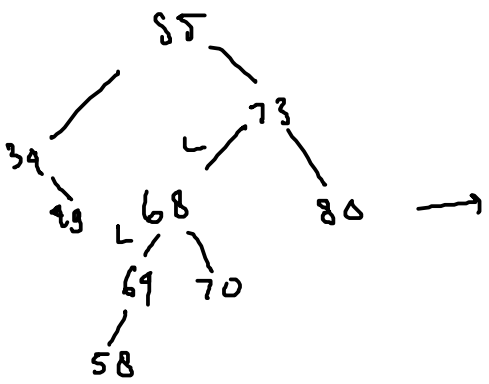


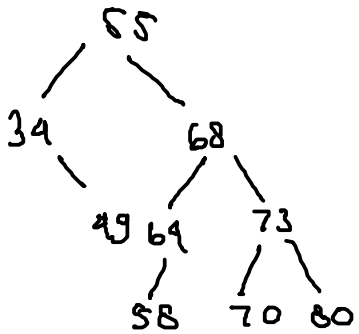
Bagian B. Esai

1. Diberikan AVL tree sebagai berikut.



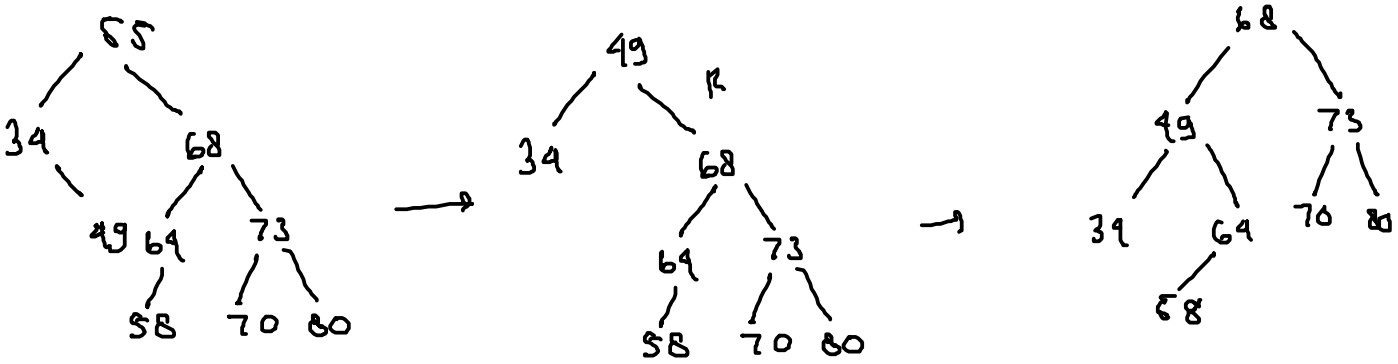
a. Lakukan simulasi proses **insert elemen 70**. Jelaskan proses rotasi yang terjadi (jika ada) dan sertakan gambar perubahan tree-nya!
Jawab:





Nama: _____ NPM: _____ Kelas: _____

b. Melanjutkan soal a (setelah insert elemen 70), lakukan simulasi proses **remove elemen 55** menggunakan kaidah **predecessor in-order**. Jelaskan proses rotasi yang terjadi (jika ada) dan sertakan gambar perubahan tree-nya!
Jawab:



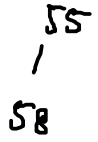
Nama: _____ NPM: _____ Kelas: _____

2. Suatu array sudah berisikan 7 data sebagai berikut.

indeks	0	1	2	3	4	5	6
data	19	58	27	46	55	34	30

Pada data tersebut akan dilakukan heapsort untuk mendapatkan data terurut secara menurun **(decending)**. Heapsort melakukan heapify kemudian melakukan selection/getMin() untuk membentuk data yang terurut secara descending.

a. Tuliskan isi array setiap iterasi dalam melakukan heapify (hints: ada 3 iterasi).



indeks	0	1	2	3	4	5	6
sebelum	19	58	27	46	55	34	30
setelah ke-1	19	58	27	46	55	34	30
setelah ke-2	19	46	27	58	55	34	30
setelah ke-3	19	46	27	58	55	34	30

b. Tuliskan isi array (bagian yang masih binary heap dan bagian yang sudah sorted) pada setiap interasi dalam melakukan operasi getMin() dan menaruhnya di bagian yang sorted (hints: ada 6 iterasi).

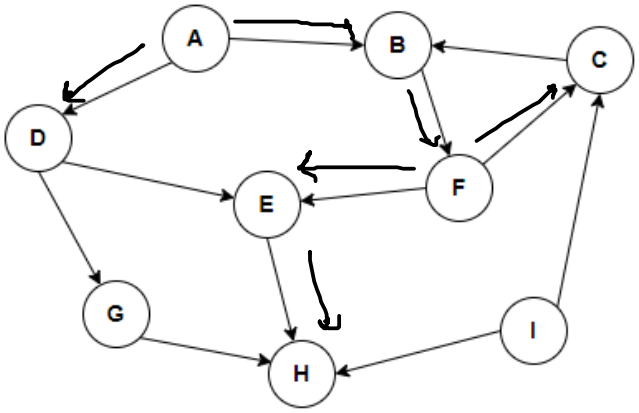
indeks	0	1	2	3	4	5	6
hasil akhir heapify tsb.	19	46	27	58	55	34	30
setelah ke-1	27	46	30	58	55	34	19
setelah ke-2	30	46	34	58	55	27	19
setelah ke-3	34	46	55	58	30	27	19
setelah ke-4	46	58	55	34	30	27	19
setelah ke-5	55	58	46	34	30	27	19
setelah ke-6	58	55	46	34	30	27	19

Nama: _____ NPM: _____ Kelas: _____

Halaman ini tidak berisi soal

Nama: _____ NPM: _____ Kelas: _____

3. Perhatikan graf berarah berikut ini, lalu jawablah pertanyaan berikut:



a. Minimum berapa kalikah DFS dijalankan supaya dapat mem-visit semua verteks pada graf di atas? Urutan pengecekan verteks yang adjacent dibebaskan. *Dengan catatan, verteks yang sudah di-visit pada DFS sebelum-sebelumnya tidak perlu di-visit ulang.*

Jawab:

2 kali DFS (A dan I)

b. Tuliskan urutan traversal masing-masing DFS yang dibutuhkan pada poin a. *Juga dengan catatan, verteks yang sudah di-visit pada DFS sebelum-sebelumnya tidak perlu di-visit ulang.*

Jawab:

DFS A :

ABFCEHDG

DFS I :

I

Nama: _____ NPM: _____ Kelas: _____

Halaman ini tidak berisi soal

Nama: _____ NPM: _____ Kelas: _____

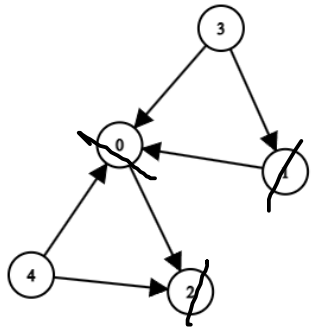
4. Perhatikan program berikut ini. Program ini adalah implementasi sebuah algoritma pada graf. Representasi graf yang digunakan adalah adjacency list, yaitu ArrayList of ArrayList.

```
import java.util.*;
class MysteryGraph {
    static void findMystery(int node, int visited[], ArrayList<ArrayList<Integer>> adj,
Stack<Integer> st) {
        visited[node] = 1;
        for(Integer it: adj.get(node)) {
            if(visited[it] == 0) {
                findMystery(it, visited, adj, st);
            }
        }
        st.push(node);
    }

    static int[] mystery(int N, ArrayList<ArrayList<Integer>> adj) {
        Stack<Integer> st = new Stack<Integer>();
        int visited[] = new int[N];

        for(int i = 0;i<N;i++) {
            if(visited[i] == 0) {
                findMystery(i, visited, adj, st);
            }
        }

        int arr[] = new int[N];
        int ind = 0;
        while(!st.isEmpty()) {
            arr[ind++] = st.pop();
        }
        return arr;
    }

    public static void main(String args[]){

        // dalam method main ini dilakukan konstruksi graf.
        ArrayList<ArrayList<Integer>> adj=new ArrayList<>();
        int n=5;
        for(int i=0; i<n; i++){
            ArrayList<Integer> arr=new ArrayList<>();
            adj.add(arr);
        }
        adj.get(4).add(0); adj.get(4).add(2);
        adj.get(3).add(0); adj.get(3).add(1);
        adj.get(1).add(0); adj.get(0).add(2);

        int res[] = mystery(5, adj);

        System.out.println("Keluaran pemanggilan mystery:" );
        for (int i = 0; i < res.length; i++) {
           System.out.print(res[i]+" ");
        }
    }
}
```

4
3
1
0
2
dt

adj[4] = 0, 2
adj[3] = 0, 1
adj[1] = 0
adj[0] = 2

Ilustrasi graf yang dibangun pada method main



Nama: _____ NPM: _____ Kelas: _____

a. Apa yang dicetak apabila program tersebut dijalankan?

9 3 1 0 2

b. Algoritma apa yang diimplementasikan pada program tersebut?

DFS

c. Berapa kompleksitas method mystery?

O(V+E)

Nama: _____ NPM: _____ Kelas: _____

5. Diberikan hash table berukuran 10 (Hsize = 10) dengan kondisi berikut ini setelah insert enam data pertama:

index	0	1	2	3	4	5	6	7	8	9
data		32	42	12		15	35			65
isActive	F	T	T	F	F	F	T	F	F	T

Atribut **isActive** digunakan untuk menandai penghapusan data dengan kaidah **lazy deletion**. True (T) berarti masih ada datanya, sedangkan False (F) berarti ada datanya namun berstatus dihapus. Fungsi hash yang digunakan adalah ***h(x) = x mod Hsize***, dengan x merepresentasikan nilai data yang disimpan.

a. Menurut Anda, pendekatan **collision resolution** apa yang diterapkan pada hash table tersebut? Berikan alasan singkat!

Quadratic Probbing

karena konz

b. Berdasarkan jawaban (a), berikan perkiraan **urutan insert keenam data** pada hash table tersebut! Data mana sajakah yang mengalami **collision** ketika di-insert?

Anjir masa semua kemungkinan ?
malas

Collision : 12, 32, 35, 65

Nama: _____ NPM: _____ Kelas: _____

c. Jika dilakukan **lazy deletion** terhadap data 12 dan 15, kemudian dilakukan **insert** data 51, bagaimana kondisi hash table tersebut sekarang? Lengkapi nilai data dan isActive setiap indeks pada array hash table berikut!

index	0	1	2	3	4	5	6	7	8	9
data		32	92	12		51	35			65
isActive	F	T	T	F	F	T	T	F	F	T