		UAS Sem. I 2015/2016 CSG2A3 (Algoritma dan Struktur Data) Kamis, 15 Desember 2015 Pk. 15.45 – 17.45 (120 menit) DNH, RSM, FSV, ADF, DQU, UNW, BBD		
<p align="center"> = Ujian ini bersifat tutup buku, no electronic device= = Kerjakan Sendiri dengan Baik , Teliti, dan Rapi = = Mulailah dengan berdoa = </p>				
Kerjakan soal langsung pada kertas ujian ini. Tidak disediakan kertas tambahan. Untuk perhitungan, gunakan bagian yang kosong di lembar nomor soal yang bersangkutan, tidak pindah ke lembar kertas nomor lain.				
Nama Mahasiswa:	NIM:	Kls:	Ruang:	Nilai (Diisi Dosen):
Salinlah pernyataan berikut: <i>Saya tidak melakukan kecurangan dalam ujian ini. Jika saya melakukan pelanggaran, maka saya bersedia diberi nilai E untuk semua Mata Kuliah pada Semester ini</i>			Tanda Tangan Mahasiswa:	

KOMPETENSI 1 : STACK

Mahasiswa mampu memahami dan mengimplementasikan model struktur data Tumpukan

SOAL 1 :

Diketahui definisi struktur data Stack yang menyimpan info berupa integer adalah sebagai berikut:

<u>{Representasi Array}</u> <u>Type</u> infotype : <u>integer</u> <u>Type</u> Stack: < info: <u>array</u> [1..50] <u>of</u> infotype Top : <u>integer</u> >	<u>{Representasi Linked List}</u> <u>Type</u> infotype : <u>integer</u> <u>Type</u> address : <u>pointer to</u> ElmStack <u>Type</u> ElmStack: < info : infotype next : address > <u>Type</u> Stack: < Top: address >
--	--

Dengan menggunakan salah satu representasi di atas (pilih salah satu), buatlah implementasi (a) prosedur Push; (b) prosedur Pop; (c) prosedur Merge!

Representasi yang dipilih:

Nama Mahasiswa:	NIM:	Kls:	Ruang:	Nilai (Diisi Dosen):
---------------------------------	----------------------	----------------------	------------------------	-----------------------------

a. **[5 poin]** Prosedur Push

Procedure Push (input/output S:Stack, input E:infotype)

{ Menambahkan sebuah elemen baru E, yang diketahui informasinya, sebagai TOP. TOP berubah nilainya.

IS: Stack mungkin kosong. E terdefinisi. Alokasi elemen baru selalu berhasil. Stack tidak penuh (khusus untuk representasi array).

FS: TOP berisi E. }

Kamus

Algoritma

b. **[5 poin]** Prosedur Pop

Procedure Pop (input/output S:Stack, output E:infotype)

{ Menghapus elemen Stack. Stack tidak boleh kosong dan mungkin menjadi kosong.

IS: Stack tidak kosong.

FS: Elemen TOP disimpan pada E. }

Kamus

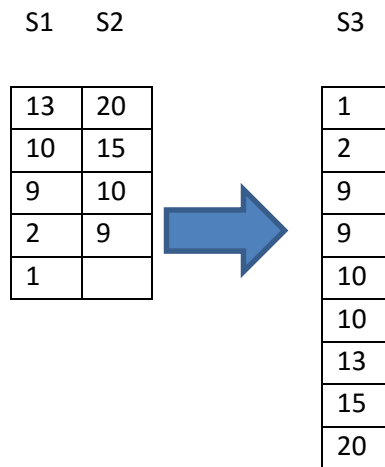
Algoritma

Nama Mahasiswa:	NIM:	Kls:	Ruang:	Nilai (Diisi Dosen):
.....	

c. **[15 poin]** Prosedur Merge

Diketahui dua buah Stack S1 dan S2 yang berisi bilangan integer yang terurut secara descending. Dengan menggunakan prosedur Push dan Pop pada poin (a) dan (b), gabungkan isi S1 dan S2 menjadi sebuah Stack baru S3 sehingga berisi bilangan integer yang terurut secara ascending.

Ilustrasi:



Procedure Merge (input S1, S2: Stack, output S3: Stack

{ IS: S1 dan S2 tidak kosong. S3 sudah terdefinisi.

FS: S1 dan S2 menjadi kosong. S3 berisi bilangan integer yang merupakan gabungan bilangan integer yang berasal dari S1 dan S2, terurut secara ascending. }

Kamus

Algoritma

Nama Mahasiswa:	NIM:	Kls:	Ruang:	Nilai (Diisi Dosen):
.....	

KOMPETENSI 2 : Queue

Mahasiswa mampu memahami dan mengimplementasikan model struktur data Antrian

SOAL 2 :

[5 poin]

Buatlah ADT Struktur Data Queue menggunakan ARRAY dengan ukuran maksimum 5 dengan infotype Queue adalah angka

Jawab :

[5 poin]

Buatlah fungsi/prosedur Add pada Queue menggunakan skema **Mekanisme Alternative 3** (circular)

Prosedur add (**input/output:** Q1 : Queue, x : integer)

Kamus

Algoritma

Nama Mahasiswa:	NIM:	Kls:	Ruang:	Nilai (Diisi Dosen):
.....	

[15 poin]

Tuliskan isi dari masing-masing Queue berikut jika dilakukan serangkaian operasi add dan delete berikut sesuai dengan alternative mekanisme operasi queue I (tail yang bergerak), II (head dan tail bergerak), dan III (circular)

Operasi	Mekanisme 1	Mekanisme 2	Mekanisme 3
START	<div>head 0 tail 0 </div> <div> <div></div><div></div><div></div><div></div><div></div> <div>1 2 3 4 5</div> </div>	<div>head 0 tail 0 </div> <div> <div></div><div></div><div></div><div></div><div></div> <div>1 2 3 4 5</div> </div>	<div>head 0 tail 0 </div> <div> <div></div><div></div><div></div><div></div><div></div> <div>1 2 3 4 5</div> </div>
Add(4)	<div>head tail </div> <div> <div></div><div></div><div></div><div></div><div></div> <div>1 2 3 4 5</div> </div>	<div>head tail </div> <div> <div></div><div></div><div></div><div></div><div></div> <div>1 2 3 4 5</div> </div>	<div>head tail </div> <div> <div></div><div></div><div></div><div></div><div></div> <div>1 2 3 4 5</div> </div>
Add(2)	<div>head tail </div> <div> <div></div><div></div><div></div><div></div><div></div> <div>1 2 3 4 5</div> </div>	<div>head tail </div> <div> <div></div><div></div><div></div><div></div><div></div> <div>1 2 3 4 5</div> </div>	<div>head tail </div> <div> <div></div><div></div><div></div><div></div><div></div> <div>1 2 3 4 5</div> </div>
Delete	<div>head tail </div> <div> <div></div><div></div><div></div><div></div><div></div> <div>1 2 3 4 5</div> </div>	<div>head tail </div> <div> <div></div><div></div><div></div><div></div><div></div> <div>1 2 3 4 5</div> </div>	<div>head tail </div> <div> <div></div><div></div><div></div><div></div><div></div> <div>1 2 3 4 5</div> </div>
Add(6)	<div>head tail </div> <div> <div></div><div></div><div></div><div></div><div></div> <div>1 2 3 4 5</div> </div>	<div>head tail </div> <div> <div></div><div></div><div></div><div></div><div></div> <div>1 2 3 4 5</div> </div>	<div>head tail </div> <div> <div></div><div></div><div></div><div></div><div></div> <div>1 2 3 4 5</div> </div>
Add(8)	<div>head tail </div> <div> <div></div><div></div><div></div><div></div><div></div> <div>1 2 3 4 5</div> </div>	<div>head tail </div> <div> <div></div><div></div><div></div><div></div><div></div> <div>1 2 3 4 5</div> </div>	<div>head tail </div> <div> <div></div><div></div><div></div><div></div><div></div> <div>1 2 3 4 5</div> </div>
Delete	<div>head tail </div> <div> <div></div><div></div><div></div><div></div><div></div> <div>1 2 3 4 5</div> </div>	<div>head tail </div> <div> <div></div><div></div><div></div><div></div><div></div> <div>1 2 3 4 5</div> </div>	<div>head tail </div> <div> <div></div><div></div><div></div><div></div><div></div> <div>1 2 3 4 5</div> </div>
Add(2)	<div>head tail </div> <div> <div></div><div></div><div></div><div></div><div></div> <div>1 2 3 4 5</div> </div>	<div>head tail </div> <div> <div></div><div></div><div></div><div></div><div></div> <div>1 2 3 4 5</div> </div>	<div>head tail </div> <div> <div></div><div></div><div></div><div></div><div></div> <div>1 2 3 4 5</div> </div>
Delete	<div>head tail </div> <div> <div></div><div></div><div></div><div></div><div></div> <div>1 2 3 4 5</div> </div>	<div>head tail </div> <div> <div></div><div></div><div></div><div></div><div></div> <div>1 2 3 4 5</div> </div>	<div>head tail </div> <div> <div></div><div></div><div></div><div></div><div></div> <div>1 2 3 4 5</div> </div>

Nama Mahasiswa:	NIM:	Kls:	Ruang:	Nilai (Diisi Dosen):
---------------------------------	----------------------	----------------------	------------------------	-----------------------------

Add(7)	head <input type="text"/> tail <input type="text"/>	head <input type="text"/> tail <input type="text"/>	head <input type="text"/> tail <input type="text"/>
	<input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/> 1 2 3 4 5	<input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/> 1 2 3 4 5	<input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/> 1 2 3 4 5
Add(1)	head <input type="text"/> tail <input type="text"/>	head <input type="text"/> tail <input type="text"/>	head <input type="text"/> tail <input type="text"/>
	<input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/> 1 2 3 4 5	<input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/> 1 2 3 4 5	<input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/> 1 2 3 4 5

KOMPETENSI 3 : Tree

Mahasiswa mampu memahami dan mengimplementasikan model struktur data Pohon

SOAL 3 :

[5 poin]

Diasumsikan terdapat T merupakan Binary Search Tree (BST) di mana setiap node dalam T menyimpan nilai integer. Buatlah representasi ADT dari BST tersebut menggunakan pointer

Jawab :

[5 poin]

Gambarkan ilustrasi hasil pohon BST dari T jika dilakukan proses insert node dengan urutan input sebagai berikut :

Input : 13, 4, 2, 8, 28, 16, 5, 22, 9, 14, 20, 7, 25

Jawab :

Nama Mahasiswa:	NIM:	Kls:	Ruang:	Nilai (Diisi Dosen):
.....	

[10 poin]

Tuliskan hasil penelusuran traversal pada BST tersebut secara inorder, preorder, postorder, dan levelorder

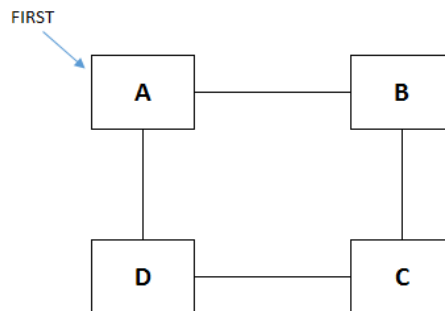
Jawab :

KOMPETENSI 4 : Graph

Mahasiswa mampu memahami dan mengimplementasikan model struktur data Graf

SOAL 4 :

Terdapat sebuah graf sebagai berikut :



Adapun deklarasi struktur data Graf diatas dalam bentuk Multi linked List sebagai berikut :

```
Type Vertex : < info : char,
                nextVertex : pointer to Vertex
                nextEdge : pointer to Edge>
Type Edge : < info : char,
              Next : pointer to Edge>
Type Graf : < first : pointer to Vertex>
```

a. [5 poin] Buatlah **gambar** representasi multi linked list untuk Graf diatas

Jawab :

Nama Mahasiswa:	NIM:	Kls:	Ruang:	Nilai (Diisi Dosen):
.....	

- b. [5 poin] Gambarkan perubahan dari Multi Linked List pada soal 3.a jika operasi-operasi ini dilakukan :
Note : sifat operasi saling terkait (efek domino)

- Buatlah agar Vertex A dan C menjadi tetangga satu sama lain

Jawab :

- Hapuslah Vertex D

Jawab :

- c. [5 poin] Buatlah Algoritma untuk menghubungkan dua buah vertex pada Graf diatas :

Procedure Connected (Input/Output G : Graf, Input : infoVertex1, infoVertex2: char)

{I.S terdapat Sebuah Graf G. Terdapat dua buah info vertex yang tidak terhubung yaitu infoVertex1 dan infoVertex2.

F.S Dua buah vertex tersebut sudah terhubung}

Kamus

Function FindVertex (G : Graf, infoVertex : char) → pointer to vertex

//Function ini akan mengembalikan pointer yang menyimpan alamat dari vertex yang dicari

Procedure InsertLastEdge (Input/Output G : Graf, Input V : Vertex, Input infoEdge : char)

//Procedure ini akan menyisipkan Edge di posisi paling akhri dari Vertex V.

Algoritma

Nama Mahasiswa:	NIM:	Kls:	Ruang:	Nilai (Diisi Dosen):
.....	

d. [10 poin] Buatlah Algoritma untuk menghapus satu buah vertex pada Graf diatas :

Procedure DeleteVertex (Input/Output G : Graf, Input : infoVertex : char)
{I.S terdapat Sebuah Graf G yang berisi vertex-vertex. Asumsi vertex yang akan didelete pasti ditemukan pada Graf G.
F.S vertex yang didelete sudah tidak menjadi bagian dari Graf G dan semua vertex yang bertetanggaan denga vertex yang dihapus tidak lagi saling terhubung}

Kamus
Function FindVertex (G : Graf, infoVertex : char)↗ pointer to Vertex
//Function ini akan mengembalikan pointer yang menyimpan lokasi vertex yang dicari

Procedure DeleteEdge (Input/Output G : Graf, Input E : pointer To Edge)
//Procedure ini akan menghapus Edge yang ditunjuk oleh pointer E)

Procedure DeleteVertex (Input/Output G : Graf, Input V : pointer to Vertex)
//Procedure ini akan menghapus Vertex yang ditunjuk oleh pointer V)

Algoritma

Nama Mahasiswa:	NIM:	Kls:	Ruang:	Nilai (Diisi Dosen):
.....	

KOMPETENSI 5 : Reskursif

Mahasiswa mampu memahami dan mengimplementasikan fungsi rekursif

SOAL 5 :

Gunakan deklarasi BST di nomor sebelumnya. Buat fungsi rekursif di bawah ini.

- a. **[nilai 10]** Fungsi untuk menghitung tinggi suatu pohon

function height (.....) → integer
{fungsi menghitung dan mengembalikan menghasilkan tinggi sebuah pohon }

Kamus

Algoritma

- b. **[nilai 10]** Fungsi untuk mengecek apakah sebuah pohon seimbang atau tidak. Pohon seimbang jika perbedaan tinggi cabang kiri dan kanan lebih dari 1

function isBalancedTree (.....) → boolean
{ Fungsi menghasilkan true jika pohon seimbang dan false jika tidak seimbang. Pohon tidak kosong }

Kamus

function height (.....) → integer {pakai fungsi pada point a}
{menghasilkan tinggi pohon}

Algoritma

Nama Mahasiswa:	NIM:	Kls:	Ruang:	Nilai (Diisi Dosen):
.....	