

Kumpulan QUIZ Algoritma & Struktur Data 2009

Dr. Anto Satriyo Nugroho, M.Eng

1. Jelaskan apa yang dimaksud dengan ALGORITMA & STRUKTUR DATA
2. Jelaskan ungkapan terkenal dari Wirth berikut:

Program = Algoritma + Struktur Data

3. BUBBLE SORT

- a) Urutkan deretan bilangan berikut dengan bubble sort sedemikian hingga nilai terbesar berada di paling awal

80 84 100 24 79 85 91

- b) Berapakah maksimal langkah yang diperlukan untuk mengurutkan n bilangan memakai algoritma tsb. ?
- c) Apakah yang anda peroleh setelah sebuah iterasi pada bubble sort selesai ?
- d) Lengkapilah program bubble sort berikut dan jelaskan langkah demi langkahnya

```
Void bubble_sort(int x[], int datanum)
```

```
{
```

```
    int i,j;
```

```
    int tmp;
```

```
        ?
```

```
}
```

4. Complexity

- a) Hitunglah complexity algoritma bubble sort !
- b) Sederhanakan notasi berikut:

$$O(3n^2 + 2^n + 5n) + O(100n \log n + 0.3n) + O(3n^3) =$$

- c) Sederhanakan notasi berikut

$$O(2n^2 + 3n \log n) \times O(0.4n) =$$

5. Tuliskan code untuk alokasi memory (memakai fungsi **malloc()**), untuk menampung array **y** dengan tipe data float, sebanyak **N**. Tuliskan juga pengecekan error, untuk mengantisipasi apabila alokasi memory tsb. gagal

6. Jelaskan apakah yang disebut

- a) Linear List
- b) Stack
- c) Queue

7. STACK: Reverse Polish Notation

a) Ubahlah perhitungan di bawah ke dalam RPN

$$\frac{\frac{20}{5} * \left(\frac{104}{4} - \frac{15}{3} \right)}{\frac{63}{7} - 2} - 5$$

b) Implementasikan perhitungan di atas dengan stack, dan gambarkan kondisi stack tiap tahap

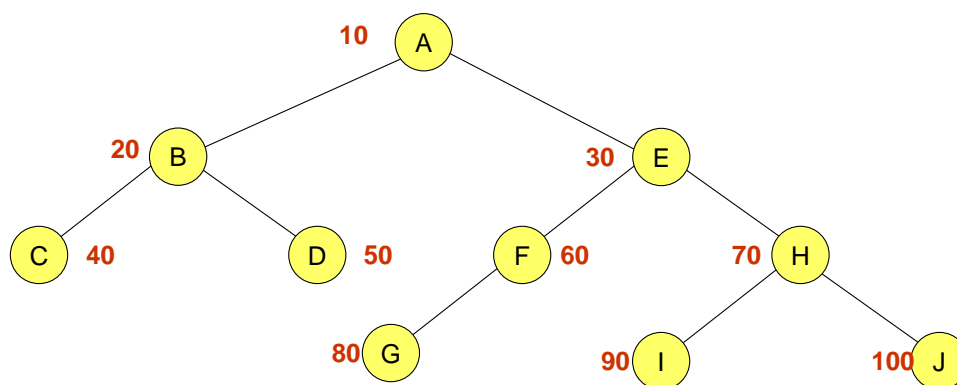
8. QUEUE

a) Jelaskan teknik implementasi queue memakai RING BUFFER.

b) Masalah yang dihadapi saat mengimplementasikan queue memakai RING BUFFER adalah saat *front* (pointer yang menunjuk ke awal queue) dan *rear* (pointer yang menunjuk ke bagian paling belakang dari queue) menunjuk ke tempat yang sama ($front == rear$), kita tidak bisa membedakan apakah queue itu kosong atau penuh. Jelaskan cara mengatasi masalah tsb. ?

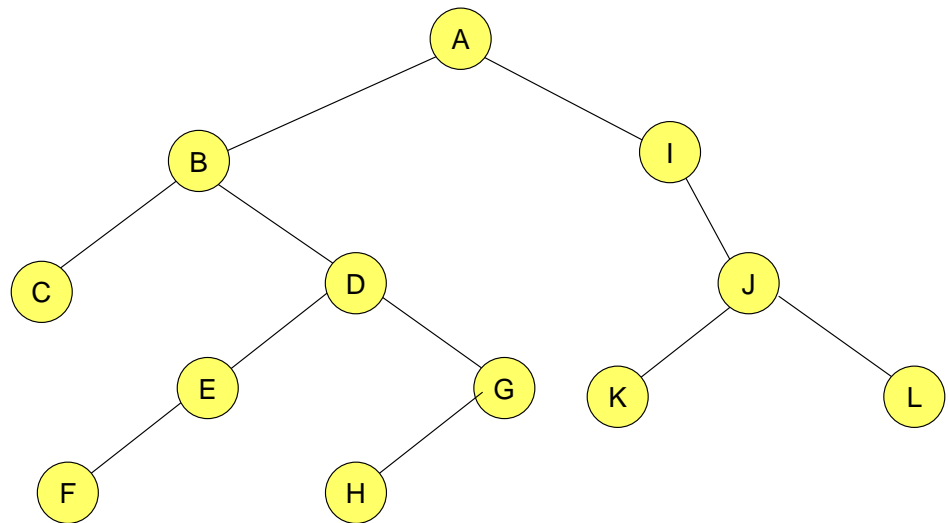
9. Jelaskan teknik implementasi Stack memakai linked-list. Apakah kelebihan yang diperoleh jika stack tersebut diimplementasikan memakai linked list daripada memakai array ?

10. Gambarkan implementasi dari binary tree berikut

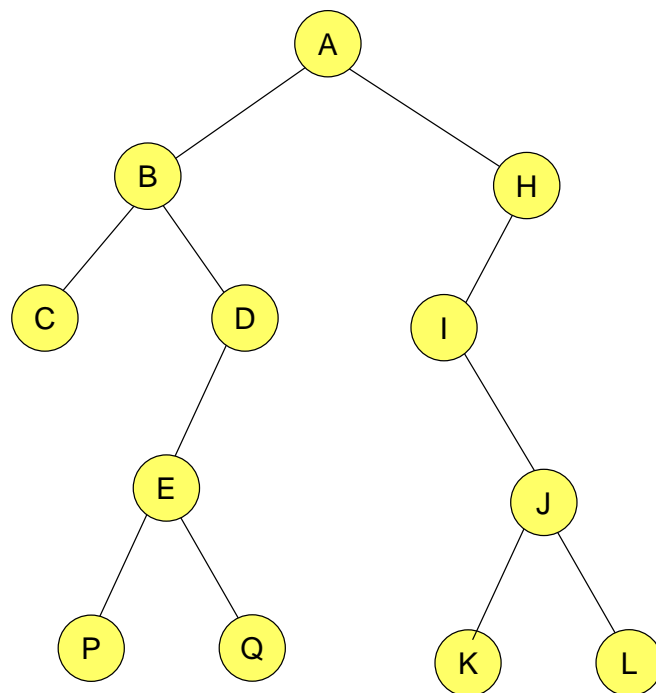


11. Tuliskan hasil PREORDER, INORDER dan POSTORDER traversal untuk binary tree berikut.

(a)

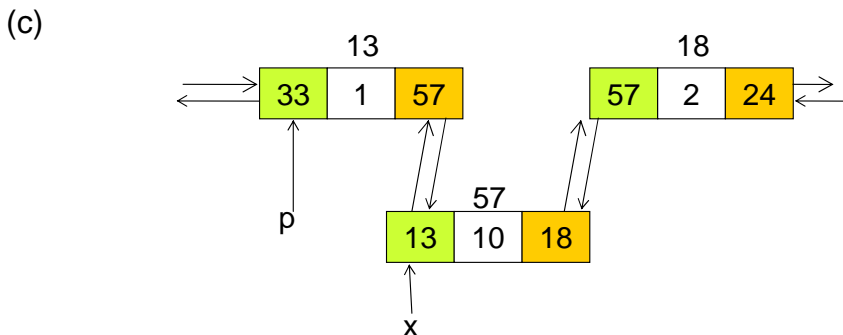
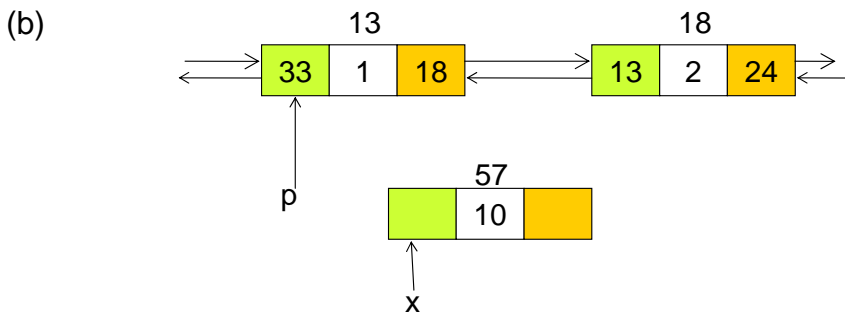
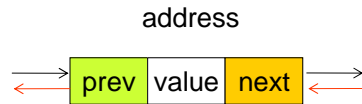


(b)

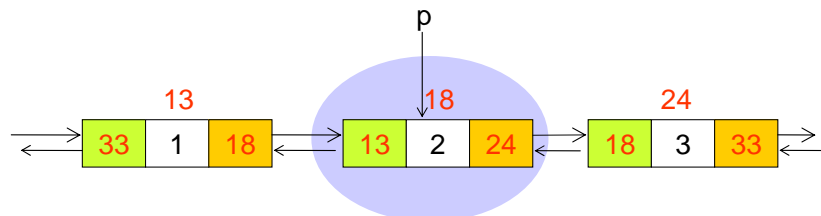


12. Pada struktur data doubly-linked list, tuliskan dalam bahasa C dalam EMPAT (4) baris, cara menambahkan sebuah sel baru yang ditunjuk oleh pointer x, sesudah suatu sel yang ditunjuk oleh pointer p. Untuk memudahkan penulisan, perhatikan bagan di bawah (a) struktur data yang dipakai (b) kondisi list sebelum sel baru ditambahkan (b) kondisi list sesudah sel baru ditambahkan

```
(a) struct CELL {
    struct CELL *prev;
    struct CELL *next;
    int value;
};
```



13. Pada struktur data doubly linked list, tuliskan pseudocode dalam 3 baris cara menghapus sebuah CELL yang ditunjuk oleh pointer p sebagaimana gambar berikut



14. Tuliskan hasil dari program berikut ! Jelaskan juga mengapa hasilnya demikian.

```
#include <stdio.h>

int main(void)
{
    int    x,y;
    int    *pt;

    x = 123;
    pt = &x;
    y = *pt;

    printf("x=%d *pt= %d y=%d \n",x, *pt, y);

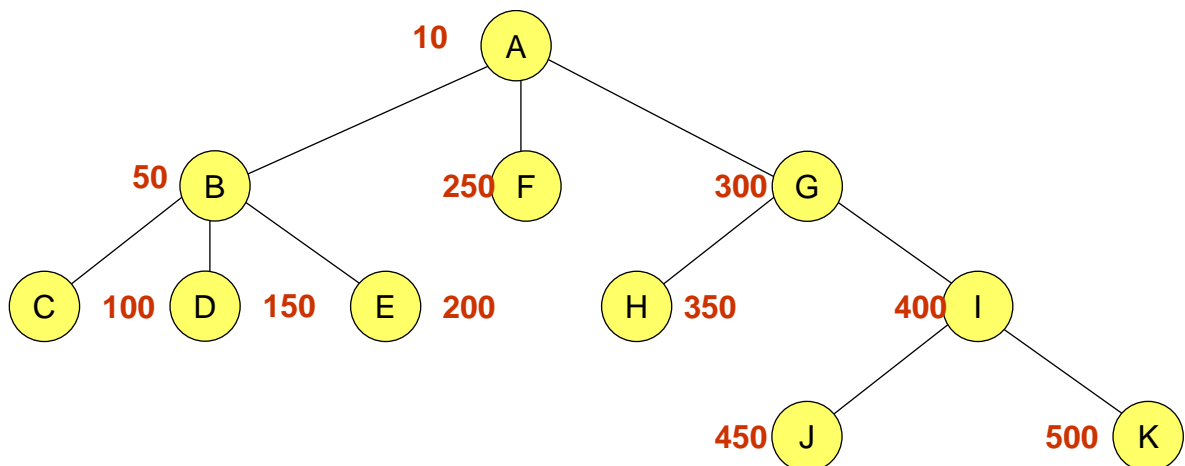
    pt = &y;

    *pt = 999;

    printf("x=%d *pt= %d y=%d \n",x, *pt, y);
}
```

15. Jelaskan implementasi tree berikut

- memakai linked-list
- memakai binary tree



16. Apakah yang dimaksud dengan recursive function ? Tuliskan fungsi untuk menghitung factorial memakai bentuk recursive.
17. Jelaskan teknik pencarian data pada linear search
18. Jelaskan teknik pencarian data pada binary search
19. Jelaskan teknik penambahan data pada binary search tree
20. Jelaskan teknik penghapusan data pada binary search tree
21. Sorting

Diketahui deretan angka sbb.

80 84 100 24 79 85 91 65 17 3 1 21

- a) Urutkan data tsb. memakai Selection Sort, agar elemen terkecil berada paling depan (urutan pertama), semakin ke belakang semakin besar
- b) Urutkan data tsb. memakai Selection Sort, agar elemen terbesar berada paling depan (urutan pertama), semakin ke belakang semakin kecil
- c) Urutkan data tsb. memakai Insertion Sort, agar elemen terkecil berada paling depan (urutan pertama), semakin ke belakang semakin besar
- d) Urutkan data tsb. memakai Insertion Sort, agar elemen terbesar berada paling depan (urutan pertama), semakin ke belakang semakin kecil

22.