

DEMO PRAKTIKUM 3

Nama : Wildan Fauzy M. H.

NRP : 5025221044

AC Praktikum : 0

AC Revisi : 4 (Dayat8, _DL, RPP, ERSY3)

Dayat with Adelson-Velsky and Landis

Time limit: 1s

Memory limit: 256MB

Ding liren

Dayat8

Rahasia Prajurit Perang

Time limit: 1s

Memory limit: 256MB

Senin Ersya

1. Pembahasan Soal Dayat :

Soal dayat pada program berikut menggunakan template code untuk membuat AVL Tree yang dapat diakses pada Github Struktur Data Teknik Informatika ITS. Adapun fungsi di luar template code yakni :

```
void dfs(AVLNode *root, int col, map<int, vector<ull>> &columnMap) //masukin avl ke indeks map
{
    if (root == nullptr)
    {
        return;
    }

    columnMap[col].push_back(root->data);

    dfs(root->left, col - 1, columnMap);
    dfs(root->right, col + 1, columnMap);
}
```

```
void VerticalPrint(AVLNode *root) //Vertical print yang akan mengeluarkan nilai tiap indeks
{
    ull jumlah = 0, final = 0;
    if (root == nullptr) //kosong
    {
        cout << 0;
        return;
    }

    std::map<int, std::vector<ull>> columnMap; //deklarasi map memakai vektor agar salah satu kolom bisa nyimpen lebih dari satu
    dfs(root, 0, columnMap); //nyimpen avl ke map

    for (const auto &column : columnMap) //iterasi map
    {
        // std::cout << "Kolom " << column.first << ":\n";
        for (int data : column.second) //akses kolom vektor kedua
        {
            jumlah += data;
        }
        final += jumlah * jumlah;
        jumlah = 0;
    }
    cout << final;
}
```

Penjelasan tiap function terletak pada ss Code function masing-masing.

Adapun fungsi utamanya adalah :

```
int main()
{
    AVL Tree;
    avl_init(&Tree);
    int kunci;

    string kartu;
    while (cin >> kartu)
    {
        if (kartu == "Insert")
        {
            cin >> kunci;
            avl_insert(&Tree, kunci);
        }

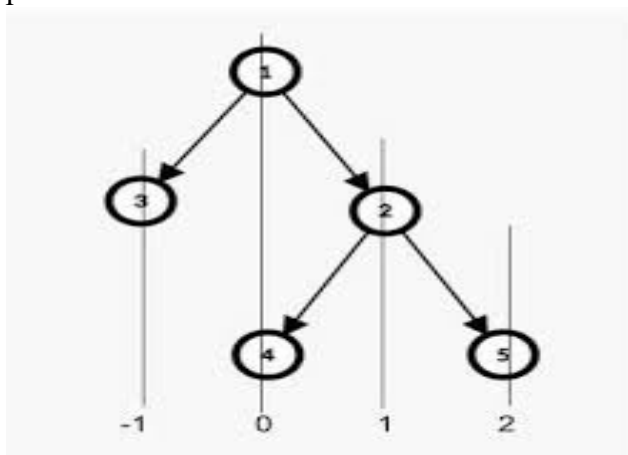
        else
        {
            VerticalPrint(Tree._root);
            cout << endl;
        }
    }
    return 0;
}
```

Pada dasarnya, Algoritma dari program ini adalah :

1. User akan memasukkan perintah yakni Insert atau password
2. Apabila perintah yang dimasukkan adalah Insert, maka angka setelah perintah akan dimasukkan ke dalam AVL Tree.
3. Apabila perintah user adalah password, maka program akan memanggil fungsi "VerticalPrint" dan mengeluarkan hasil yang dikeluarkan dari fungsi tersebut.

Notes :

Fungsi "VerticalPrint" ditunjang oleh fungsi dfs yang akan memetakan tree seperti pada ilustrasi berikut :



Kemudian, pada fungsi “VertialPrint”, tree akan dipetakan dengan menggunakan vektor yang akan mengklasifikasikan denah masing-masing angka dan kemudian akan diberi hitungan sesuai dengan yang diminta pada soal.

2. Penjelasan Soal Ding Liren :

Soal DingLiren dapat dipecahkan dengan menggunakan program berikut :

```
#include <iostream>
#include <string>
#include <unordered_map>

using namespace std;

int main() {
    int n;
    cin >> n;

    unordered_map<string, int> freq; //Map untuk cek ada angka kembar atau gak
    string x;

    for (int i = 0; i < n; i++)
    {
        string num;
        cin >> num;
        freq[num]++;
        if (freq[num] == 2) //ada angka yang sama
        {
            x = num; // angka kembar yang akan direversed
        }
    }

    string rev_x = ""; //deklarasi string kosong

    for (int i = x.length()-1; i >= 0; i--) //reversed angka dengan cara rewrite dari belakang
    {
        rev_x += x[i];
    }

    string sum = ""; //deklarasi string kosong+
    int carry = 0;
    for (int i = 0; i < x.length(); i++) //penjumlahan string
    {
        int digit_x = x[i] - '0';
        int digit_rev_x = rev_x[i] - '0';
        int digit_sum = digit_x + digit_rev_x + carry;
        carry = digit_sum / 10;
        digit_sum %= 10;
        sum += to_string(digit_sum);
    }

    if (carry > 0)
    {
        sum += to_string(carry);
    }

    string result = "";
    for (int i = sum.length()-1; i >= 0; i--)
    {
        result += sum[i];
    }

    cout << result << endl;

    return 0;
}
```

Adapun algoritma yang diterapkan pada Program ini adalah :

1. User memasukkan jumlah angka yang akan dimasukkan dan memasukkan angka lainnya sebanyak n jumlah.

2. Program mendeklarasikan suatu variabel `x` untuk menyimpan angka yang akan keluar dua kali dan disokong dengan `'Unordered_map'` untuk menyimpan frekuensi suatu angka dimasukkan.
3. Ketika angka kembar ditemukan, angka tersebut juga akan dimasukkan ke variabel `'rev_x'` dengan cara loop for yang membalikkan angka pada variabel `x` dan loop dimulai dari indeks terakhir yakni `'x.length - 1'` sampai 0.
4. Kemudian penambahan kedua variabel tersebut akan dilakukan dengan menggunakan loop for. Pada setiap iterasi, angka-angka pada kedua string tersebut akan diambil, diubah menjadi angka integer, dan kemudian dijumlahkan. Jika hasil penjumlahan lebih dari 10, maka nilai carry akan diatur dan hasil penjumlahan akan di-modulus dengan 10 untuk mendapatkan digit yang tepat. Digit hasil penjumlahan tersebut akan ditambahkan ke dalam string `'sum'` yang sudah dideklarasikan sebelumnya. setelah loop for selesai dan nilai carry masih lebih dari 0, maka digit terakhir carry akan ditambahkan ke dalam string `sum`.
5. Terakhir, akan digunakan loop for yang akan membalik string `sum` dan mengisinya ke dalam variabel `result`. Loop dimulai dari indeks terakhir `sum.length() - 1` hingga indeks ke-0. Setelahnya, hasil di outputkan.

3. Pembahasan Soal Rahasia Prajurit Perang

Berikut adalah program yang digunakan untuk menyelesaikan soal RPP :

```
#include <iostream>
#include <map>
#include <string>
using namespace std;

struct Prajurit // truct untuk menyimpan data prajurit
{
    int usia;
    string asal;
};

int main()
{
    int n, q;
    cin >> n;

    map<string, Prajurit> data;
    for (int i = 0; i < n; i++)
    {
        string nama;
        Prajurit prajurit;
        string temp;
        cin >> temp >> nama >> temp >> prajurit.usia >> temp >> prajurit.asal;
        data[nama] = prajurit;
    }

    cin >> q;
    for (int i = 0; i < q; i++)
    {
        string nama;
        cin >> nama;
        Prajurit prajurit = data[nama];
        cout << "Usia: " << prajurit.usia << endl;
        cout << "Asal: " << prajurit.asal << endl;
    }
    return 0;
}
```

Adapun algoritmanya adalah :

1. User memasukkan jumlah prajurit yang akan didata
2. Sebuah map dengan nama 'data' akan dideklarasikan dengan string sebagai key dan struct prajurit sebagai value. Penyimpanan data menggunakan map dilakukan agar pencarian berdasarkan nama dapat dilakukan dengan cepat dan efisien.
3. Dilakukan Loop For sebanyak n kali untuk menginput data prajurit seperti nama, usia, dan asal.
4. User memasukkan jumlah data yang akan dicari melalui variabel q.
5. Loop for akan dilakukan sebanyak q kali untuk mencari dan menampilkan data prajurit.

4. Penjelasan Soal Senin Ersya

Berikut adalah program yang dapat menyelesaikan Soal Senin Ersya :

```
#include <iostream>
#include <string>
#include <map>

using namespace std;

int main()
{
    int t;
    cin >> t;

    map<int, int> tumpukan;

    while (t--)
    {
        string perintah;
        int vol;

        cin >> perintah >> vol;

        if (perintah == "Taro")
        {
            tumpukan[vol]++;
        }

        else if (perintah == "Ingfo")
        {
            if (tumpukan[vol] == 0)
            {
                cout << "No Ingfo" << endl;
            }

            else
            {
                int pos = 1;
                for (auto it = tumpukan.rbegin(); it != tumpukan.rend(); it++)
                {
                    if (it->first == vol)
                    {
                        cout << "Info manga di tumpukan ke - " << pos << endl;
                        break;
                    }
                    pos += it->second;
                }
            }
        }

        else
        {
            cout << "Gajelas" << endl;
        }
    }

    return 0;
}
```

Adapun Algoritma yang membangun program tersebut ialah :

1. User memasukkan jumlah kasus uji melalui cin >> t;.

2. Sebuah Map tumpukan dengan tipe key dan value berupa integer dideklarasikan. Map berfungsi untuk menyimpan informasi mengenai volume manga yang ada dalam tumpukan.

3. Program akan melakukan loop while sebanyak t kali untuk melakukan operasi pada tumpukan manga. Pada setiap iterasi, program membaca input perintah dan volume manga (vol) melalui cin >> perintah >> vol;.

4. Jika perintah adalah "Taro", manga dengan volume vol akan ditambahkan ke dalam tumpukan. Informasi tersebut kemudian disimpan ke dalam map tumpukan dengan menggunakan volume manga sebagai key dan nilai valuenya akan di increment kan.

5. Jika perintah adalah "Ingfo", akan dilakukan pencarian informasi tentang manga dengan volume vol dalam tumpukan. Jika volume manga tersebut tidak ada dalam tumpukan (nilai value pada map tumpukan[vol] = 0), maka program akan mengeluarkan "No Ingfo". Jika volume manga tersebut berada dalam tumpukan, program akan mencari posisi manga tersebut dalam tumpukan dengan mengiterasi map tumpukan dari belakang (rbegin() dan rend()). Di setiap iterasi, Program akan memeriksa apakah volume manga pada elemen saat ini (it->first) sama dengan vol. Jika ya, Program akan mengeluarkan posisi manga tersebut dalam tumpukan (pos) dan break dari loop. Jika tidak, Program akan menambahkan manga pada elemen saat ini ke dalam pos.

6. Jika program mendapatkan perintah selain "Taro" atau "Ingfo", Program akan mengeluarkan "Gajelas".