

Pembahasan Soal Kuis

Sekolah Olimpiade LOPI bidang Informatika 2025

Perhatikan potongan program berikut untuk nomor 36 – 37

```
int main(){
    int t;
    cin>>t;
    int x,y;
    vector<int> angka;
    while (t--){
        cin>>x>>y;
        if(x > y){
            angka.push_back(x);
            angka.push_back(y);
        }else if(x == y){
            angka.push_back(x);
        }else{
            angka.push_back(y);
            angka.push_back(x);
        }
    }
    for(int number : angka){
        cout<<number<<" ";
    }
    return 0;
}
```

36. Jika keluaran program berupa sekumpulan angka 1 0 0 3 2 4 2 masukan mengikuti format input dari program yang TIDAK mungkin di bawah ini?

input dari program

a. 4
1 0
0 0
3 2
2 4

b. 5
1 1
0 0
0 0
3 2
2 4

c. 4
1 0
2 4
3 0
2 4

0 3
2 2
2 4
d. 6
1 1
0 0
0 0
3 3
2 2
4 2

e. 5
1 1
0 0
0 0
3 2
4 2

1 0 0 3 2 4 2

1 0 0 3 2 4 2

Masukkan angka terbesar dulu ke dalam vektor

Pilihan A ⇒ (Benar)

1 0 0 3 2 4 2

4
1 0
3 2
2 4

37. Jika keluaran program menghasilkan 5 angka berbeda, berapa nilai t terbesar dan t terkecil yang dapat menjadi nilai masukan? {Tuliskan jawaban berupa nilai t terbesar dan t terkecil secara berurutan dipisahkan oleh spasi}

Jawaban: 5 3

1 2 3 4 5

1 1
2 2
3 3
4 4
5 5

} $t_{\max} = 5$

1 2
3 4
5 5

} $t_{\min} = 3$

array / Lanir \Rightarrow access by index

index = urutan - 1

Daftar angka ganjil
Daftar nama orang

size of array
 \uparrow
0 1 2 3 4

int angkaG[5] = {1, 3, 5, 7, 9}

Ke-1, index = 0

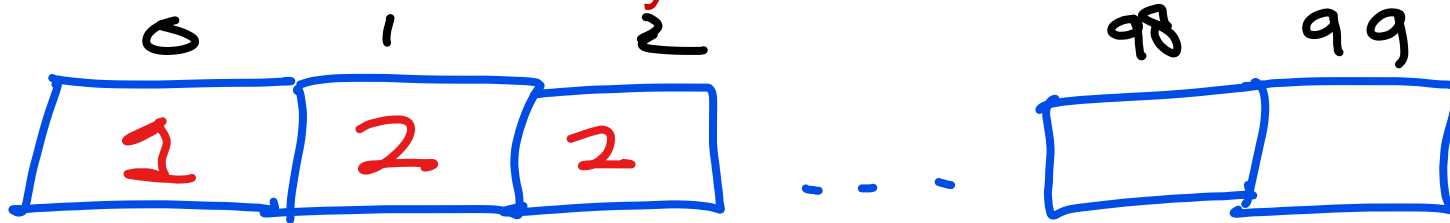
angka Pertama dalam daftar ?

\rightarrow angka G[0]

array \rightarrow tipe data derived

int rumah[100]; belum tentu 100 rumah terisi

Static rumah sebanyak 100 itu akan di-inisiasi di dalam memori



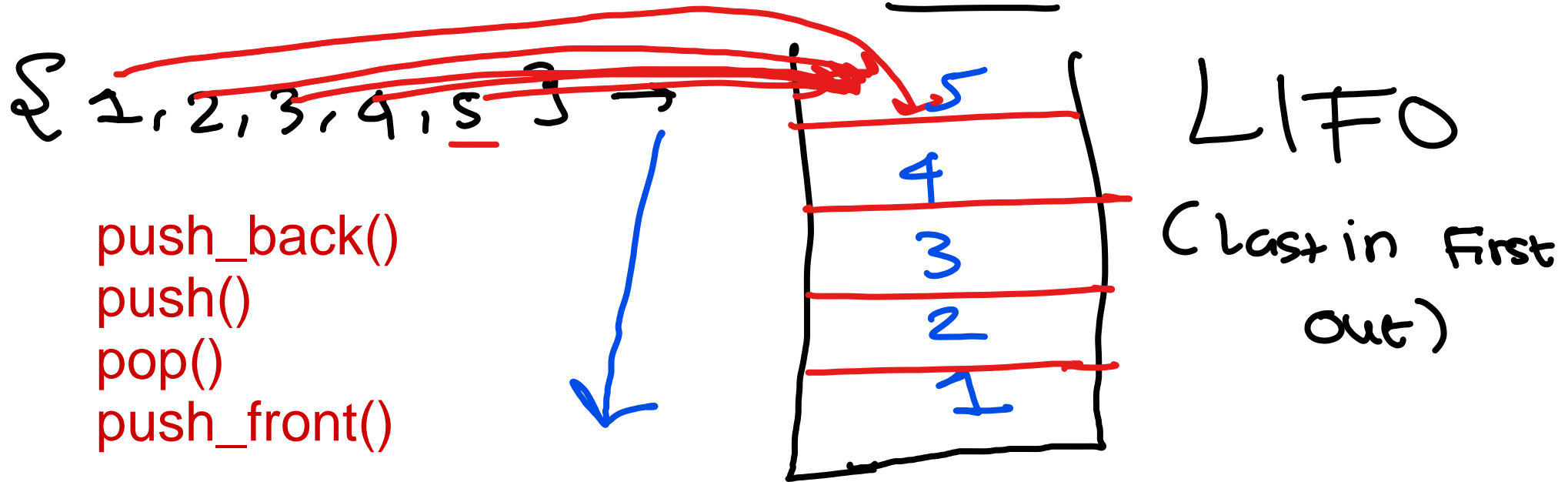
Bagaimana jika kita ingin maintain data kita supaya nanti alokasi di memorinya bisa efisien : Ukurannya dinamis

Di dalam memori ukurannya sesuai yang digunakan

Dynamic DS

⇒ Stack, Queue

* Stack (Tumpukan)

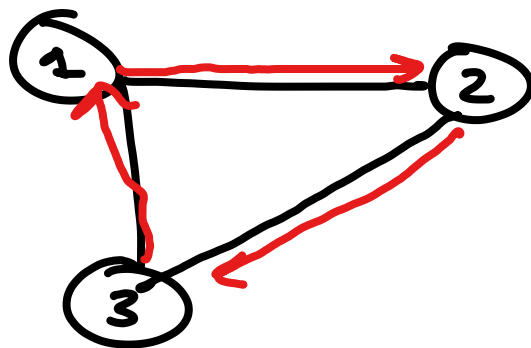


* Queue (antrian)

FIFO
(First in First out)

← 1 2 3 4 5
First Last

Graph →



Traverse

- * BFS → Queue
- * DFS → stack / Rekursif

Vector → Scalar DS ↓ → ↗
↓ Dynamic array

Vector -> Queue dan Array

