
Pembahasan Investigasi Pembangunan Gedung

Deskripsi Singkat

Terdapat 2 jenis LinkedList yang digunakan, Singly LinkedList dan Doubly LinkedList. Ada 2 jenis pointer, Denji dan Iblis. Kedua pointer memiliki 1 atau 2 query tersendiri yang tidak dimiliki oleh yang lainnya. Kedua pointer akan saling menjauh dari satu sama lain atau saling mendekati satu sama lain. Bila mereka bertemu di suatu lantai, maka pertemuan mereka akan dicatat.

Inisiasi Awal

Pada tahap inisiasi, yang harus disiapkan adalah gedung-gedung yang digunakan dan lantai-lantainya. Untuk melakukan ini, buat LinkedList untuk menyimpan semua *tower*, dan lakukan *for-loop* untuk menambahkan setiap lantai ke dalam tower. Jangan lupa untuk *add tower* yang sudah dibuat ke dalam LinkedList.

Berikut ini adalah potongan program saat inisiasi gedung.

```
for (int i = 0; i < jumlahGedung; i++){
    String namaGedung = in.next();
    int jumlahLantai = in.nextInt();
    Tower newTower = new Tower(namaGedung);

    for(int j = 0; j < jumlahLantai; j++){
        newTower.addLantai();
    }
    linkedTower.addTower(newTower);
}

public void addLantai() {
    Node newNode = new Node(this);
    if (head == null) {
        head = tail = newNode;
        head.previous = null;
    } else {
        Node temp = tail;
        tail = newNode;
        tail.previous = temp;
        temp.next = newNode;
    }
}
```

```

    }

    tail.next = null;
    size++;
}

public void addTower(Tower newTower){
    if (head == null) {
        head = tail = newTower;
        head.previous = newTower;
        tail.next = newTower;
    } else {
        Tower temp = tail;
        tail = newTower;
        tail.previous = temp;
        temp.next = newTower;
        tail.next = head;
        head.previous = tail;
    }
}
}

```

Lakukan *set pointer* untuk menempatkan kedua pointer ke dalam lantai gedung-gedung. Untuk Denji, gerak awal akan selalu ke atas, sedangkan untuk gerak iblis akan selalu ke bawah. Denji dan iblis akan menyimpan sedang berada di lantai berapa mereka saat ini dan setiap mereka naik/turun akan selalu dicatat.

Berikut ini adalah potongan program saat inisiasi pointer untuk Iblis. Sedangkan untuk Denji mirip dengan Iblis, hanya berbeda posisi awalnya di lantai pertama dari gedung pertama.

```

iblis.setFlags("DOWN");
Tower tempGedung = linkedTower.getHead();
iblis.setTower(tempGedung);
if (!gedungIblis.equals(tempGedung.toString())) {
    for (int j = 1; j < linkedTower.getSize(); j++) {
        if (gedungIblis.equals(tempGedung.next.toString())){
            iblis.setTower(tempGedung.next);
            break;
        }
        tempGedung = tempGedung.next;
    }
}

```

```

}

Node tempLantai = iblis.getTower().getHead();
iblis.setNode(tempLantai);
if(lantai != 1){
    for(int j = 1; j < lantai; j++){
        tempLantai = tempLantai.next;
        iblis.setNode(tempLantai);
        iblis.setKetinggian(iblis.getKetinggian() + 1);
    }
}
}

```

GERAK

Pada saat kedua karakter melakukan perpindahan, terdapat aturan yang sudah dijelaskan.

- Denji akan bergerak pertama kali dan melangkah sebanyak satu kali.
- Selanjutnya Iblis akan bergerak dengan melangkah sebanyak dua kali.
- Setiap selesai melakukan pergerakan, masing-masing akan mencatat ketinggian mereka saat ini serta mengecek apakah mereka saling bertemu di lantai tersebut atau tidak.

Hal yang perlu diperhatikan adalah:

- Jika berada di lantai paling atas dan mau bergerak ke atas, maka harus berpindah ke lantai paling atas di gedung setelahnya, lalu arah Bergeraknya menjadi ke bawah.
- Jika berada di lantai paling bawah dan mau bergerak ke bawah, maka harus berpindah ke lantai paling bawah di gedung setelahnya, lalu arah Bergeraknya menjadi ke atas.
- Setiap selesai bergerak, update status posisi lantai pointer saat ini.

Berikut ini adalah potongan kode untuk melakukan gerakan 1 langkah, berlaku untuk Denji dan Iblis.

```

public void move() {
    if (this.flags.equals("UP")) {
        if (this.node.next != null) {
            this.node = this.node.next;
            this.ketinggian++;
        }
        else {
            setTower(this.tower.next);
            this.setFlags("DOWN");
            this.node = this.tower.getTail();
            this.ketinggian = this.tower.getSize();
        }
    }
}

```

```

    } else {
        if (this.node.previous != null) {
            this.node = this.node.previous;
            this.ketinggian--;
        }
        else {
            setTower(this.tower.next);
            this.setFlags("UP");
            this.node = this.tower.getHead();
            this.ketinggian = 1;
        }
    }
}
}

```

Jangan lupa untuk selalu mengecek apakah Denji dan Iblis bertemu di lantai yang sama ketika mereka selesai bergerak. Berikut ini adalah potongan kode untuk mengeceknya.

```

public void cekSelantai() {
    if (iblis.getTower() == denji.getTower()
        && iblis.getKetinggian() == denji.getKetinggian()) {
        counter++;
    }
}
}

```

HANCUR

Query ini hanya dimiliki oleh Denji saja dan fungsinya adalah untuk menghancurkan tepat satu lantai yang berada dibawah Denji. Pada *query* ini perlu diperhatikan beberapa hal berikut.

- Apakah lantai di bawah Denji saat ini terdapat Iblis?
- Apakah saat ini Denji berada di lantai terbawah?

Setelah melakukan penghancuran, jangan lupa untuk **SELALU** *update* ketinggian Denji, Iblis (bila berada di gedung yang sama dan berada di atas Denji), dan Tower.

Berikut ini adalah potongan kode jika Denji menghancurkan lantai.

```
public void hancur() {
    Node tempDenji = denji.getNode();
    denji.getTower().setSize(denji.getTower().getSize() - 1);
    tempDenji.previous = tempDenji.previous.previous;

    if (tempDenji.previous == null) {
        denji.getTower().setHead(tempDenji);
    } else {
        tempDenji.previous.next = tempDenji;
    }

    // Jika Iblis berada di atas Denji di gedung yang sama
    if (iblis.getTower() == denji.getTower()
        && denji.getKetinggian() <= iblis.getKetinggian()) {
        iblis.setKetinggian(iblis.getKetinggian()-1);
    }

    denji.setKetinggian(denji.getKetinggian() - 1);
}
```

TAMBAH

Query ini hanya digunakan oleh Iblis saja. Pada saat melakukan pertambahan, Iblis hanya bisa menambahkan satu lantai saja di bawahnya. Pada *query* ini perlu diperhatikan beberapa hal berikut.

- Apakah iblis berada di lantai terbawah?
- Apakah Denji berada di gedung yang sama?

Saat berada di lantai terbawah, jangan lupa untuk *set head* kembali lantai gedung tersebut dengan lantai yang baru ditambah. Jangan lupa untuk **SELALU** *update* ketinggian Denji (bila berada di gedung yang sama dan berada di atas Denji), Iblis, dan Tower.

Berikut ini adalah potongan kode saat Iblis menambahkan lantai.

```
public void tambah() {
    Node newNode = new Node(iblis.getTower());
    Node tempIblis = iblis.getNode();

    newNode.getTower().setSize(newNode.getTower().getSize() + 1);

    if (tempIblis.previous == null){
        newNode.previous = null;
        iblis.getTower().setHead(newNode);
    }
    else {
        newNode.previous = tempIblis.previous;
        tempIblis.previous.next = newNode;
    }

    newNode.next = tempIblis;
    tempIblis.previous = newNode;

    // Jika Denji berada di atas Iblis di gedung yang sama
    if (iblis.getTower() == denji.getTower()
        && iblis.getKetinggian() <= denji.getKetinggian()
    ) {
        denji.setKetinggian(denji.getKetinggian() + 1);
    }

    iblis.setKetinggian(iblis.getKetinggian() + 1);
}
```

PINDAH

Pada *query* ini, intinya hanya memindahkan pointer Denji ke gedung sebelah sesuai dengan *flag* dia saat ini.

- Jika **UP** maka akan ditempatkan di lantai terbawah gedung selanjutnya.
- Jika **DOWN** maka akan ditempatkan di lantai teratas gedung selanjutnya.

Ketika berpindah ke gedung sebelah, Denji bisa saja bertemu dengan Iblis dan pertemuan ini perlu dicatat juga. Jangan lupa untuk selalu *update* posisi Denji setelah pindah.

Berikut ini adalah potongan kode saat Denji pindah.

```
public void pindah() {
    denji.setTower(denji.getTower().next);

    if (denji.getFlags().equals("UP")) {
        denji.setNode(denji.getTower().getHead());
        denji.setKetinggian(1);
    }
    else if (denji.getFlags().equals("DOWN")) {
        denji.setNode(denji.getTower().getTail());
        denji.setKetinggian(denji.getTower().getSize());
    }

    cekSelantai();
}
```