

Heap

Implementasi Max Heap

Pada max heap, setiap node kecuali root, memenuhi property: $A[\text{parent}(i)] \geq A[i]$. Pada bagian ini dicontohkan implementasi dari max heap.

1. Buatlah kelas MaxHeap dengan atribut-atributnya seperti gambar di bawah ini:

```
public class MaxHeap{  
    private int maxHeap[];  
    private int heapSize;  
    private int length;  
}
```

2. Tambahkan dua buah constructor:

- a. Constructor pertama dengan parameter *length*, yaitu ukuran maksimum dari max heap.

```
public MaxHeap(int length){  
    this.heapSize=0;  
    this.length=length;  
    this.maxHeap=new int[length+1];  
}
```

- b. Constructor kedua dengan parameter sebuah array bertipe int.

```
public MaxHeap(int[] arr){  
    this.heapSize=arr.length;  
    this.length=arr.length;  
    this.maxHeap=new int[length+1];  
    for(int i=0; i<length;i++){  
        this.maxHeap[i+1]=arr[i];  
    }  
}
```

3. Tambahkan method *getLeft*, *getRight*, dan *getParent* yang masing-masing berfungsi untuk mendapatkan indeks dari anak kiri, anak kanan, dan parent dari node *i* pada heap.

```
private int getLeft(int i){  
    return i<<1;  
}  
  
private int getRight(int i){  
    return (i<<1)|1;  
}  
  
private int getParent(int i){  
    return i>>1;  
}
```

Note: Pada implementasi di atas, indeks dari max heap dimulai dari 1. Oleh karena itu, untuk membuat max heap berukuran *length*, perlu disediakan sebuah array dengan ukuran *length+1*, sehingga dapat diisi sebanyak *length* buah elemen pada array tersebut (dari indeks 1,2,3...,length). Dampak lainnya adalah *insert* sebuah key hanya bisa dilakukan selama $\text{heapSize} \leq \text{length}$, dan element baru disimpan di lokasi ke-*heapSize+1*.

Tambahkan method-method berikut ini:

- a. `maxHeapify(int i)`
- b. `buildMaxHeap()`
- c. `insertKey(int key)`
- d. `increaseKey(int i, int key)`
- e. `extractMax()`

T0901-HeapSort

Implementasikanlah heapsort! Anda hanya perlu menambahkan sebuah method lagi di dalam kelas `MaxHeap` yang telah dibuat pada soal nomor 1.

Uji Heap

Buatlah method main yang dapat menangani input/output sebagai berikut.

Input terdiri dari sejumlah perintah, maksimal 100.000. Jenis perintah adalah:

- **insert x** , yang artinya memasukkan nilai x ke maximum priority queue. ($0 \leq x \leq 10^6$)
- **extract**, yang artinya mengeluarkan nilai maksimum, dan menuliskan ke output. Jika heap kosong, keluarkan "-".
- **end**, yang artinya akhir dari input. Urutkan sisa element yang ada pada heap dengan heap sort, kemudian keluarkan isinya.

Contoh Input

```
extract
insert 5
insert 6
insert 1
extract
insert 3
insert 9
extract
insert 10
end
```

Contoh Output

```
-
6
9
1 3 5 10
```