Work Sheet 6
Implementasi *Binary Heap Tree*

Struktur Data dan Algoritma
IKI10400
Semester Genap 2010/2011

Fakultas Ilmu Komputer
Universitas Indonesia

Batas waktu pengumpulan kode sumber:
Minggu, 1 Mei 2011 pukul 21.00 Waktu Server Aren

Kode sumber yang dinilai hanya yang dikumpulkan melalui Aren. Kode sumber yang dikumpulkan melalui mekanisme selain itu akan diabaikan dan dianggap tidak mengumpulkan.

Peringatan: jangan mengumpulkan pekerjaan beberapa menit menjelang batas waktu pengumpulan karena ada kemungkinan pengumpulan gagal dilakukan atau koneksi internet terputus!

Jika tidak dapat mengumpulkan WS sebelum batas waktu karena suatu atau beberapa hal khusus, mahasiswa yang bersangkutan harus melakukan langkah-langkah dalam SOP Perpanjangan Batas Waktu Pengumpulan Pekerjaan.

Binary Heap Tree

Nama berkas kode sumber : `SDA11106.java`
Batas waktu eksekusi program : 1 detik / kasus uji
Batas memori program : 16 MiB / kasus uji

Buatlah sebuah program Java yang mengimplementasikan operasi-operasi menambahkan data baru, menghapus elemen paling kecil (di *root*), dan *heapify* pada *Binary Heap Tree* dengan kriteria *parent* ≤ *child*! (Pada saat melakukan *push-down root*, jika nilai anak kiri sama dengan nilai anak kanan, pilihlah anak kiri)

Format Masukan

Masukan dibaca dari masukan standar. Masukan terdiri dari beberapa buah baris (antara 1 s.d. 30 000). Baris pertama berisi N ($0 \le N \le 1000$) buah bilangan bulat X ($|X| \le 100\ 000$) yang menyatakan bilangan-bilangan pada kondisi awal *Binary Heap Tree* yang perlu di-*heapify* jika tidak memenuhi kriteria di atas. Antar bilangan dipisahkan dengan sebuah karakter spasi. Jika $N = 0$, baris pertama merupakan baris kosong dan kondisi awal *Binary Heap Tree* kosong.

Untuk setiap baris berikutnya (dibaca sampai EOF), masing-masing baris pasti diawali sebuah perintah yang merupakan salah satu dari "`insert`" (untuk selanjutnya tanda kutip hanya untuk kejelasan) atau "`deleteMin`" (harap perhatikan huruf kapital pada perintah). Untuk perintah "`insert`", pasti hanya diikuti oleh sebuah karakter spasi dan sebuah bilangan bulat X pada baris yang sama. Sedangkan untuk perintah lainnya, hanya perintah itu yang berada pada baris yang dimaksud. Dijamin ketika perintah "`deleteMin`" dievaluasi, *Binary Heap Tree* tidak kosong.

Format Keluaran

Keluaran ditulis ke keluaran standar. Keluaran terdiri dari sebuah baris. Jika minimal terdapat sebuah elemen pada *Binary Heap Tree*, baris itu berisi semua elemen *Binary Heap Tree* pada kondisi terakhir setelah semua perintah dijalankan. Elemen-elemen ditulis secara *level-order* (mulai dari *level root* s.d. l*eaf*; pada *level* yang sama, mulai dari yang paling kiri s.d. paling kanan). Antar elemen dipisahkan oleh sebuah tanda koma.

Jika kondisi terakhir *Binary Heap Tree* kosong, tulis "`[Empty]`" pada baris pertama (perhatikan huruf '`E`' merupakan huruf kapital dan tanda kurung kotak juga ditulis).

Contoh Masukan

```
34 43 24 63 12 53 24 21 51
insert 55
insert 83
deleteMin
deleteMin
insert 13
insert 52
deleteMin
```

Contoh Keluaran

```
24,34,24,51,43,53,55,63,83,52
```

Batasan

Kelas-kelas yang sudah disediakan Java yang boleh digunakan hanyalah kelas-kelas dari *package* `java.lang` dan `java.io`. Kelas-kelas lainnya tidak boleh digunakan. Asisten akan memeriksa kode sumber Anda apakah terdapat kelas-kelas yang tidak boleh digunakan.

Kriteria Penilaian

Terdapat dua bagian penilaian, yaitu:

- penilaian penilai otomatis Aren (50%).
- penilaian *white-box review* (50%).

Komponen-komponen penilaian *white-box review* adalah sebagai berikut.

- Implementasi *heapify* pada *Binary Heap Tree* yang benar.
- Implementasi *insert* pada *Binary Heap Tree* yang benar.
- Implementasi *deleteMin* pada *Binary Heap Tree* yang benar.

Jika terdapat kelas-kelas yang tidak boleh digunakan dalam kode sumber berdasarkan batasan di atas, total nilai WS 6 adalah 0.