

Work Sheet 4
Rekursif Pada Tree

Struktur Data dan Algoritma
IKI10400
Semester Genap 2010/2011

Fakultas Ilmu Komputer
Universitas Indonesia

Batas waktu pengumpulan kode sumber:
Minggu, 10 April 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.

Pohon Kalkulator

Nama berkas kode sumber : SDA11104.java
Batas waktu eksekusi program : 0,5 detik / kasus uji
Batas memori program : 32 MiB / kasus uji

Latar Belakang

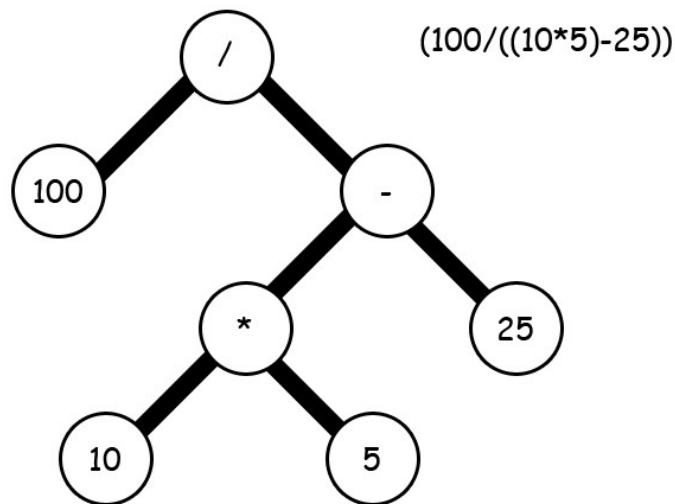
Wah, WS 4 SDA sudah keluar. Mahasiswa SDA Fasilkom UI beramai-ramai mengikuti tutorial untuk mengerjakannya. Pada saat tutorial, Moo si Pemalu telah berhasil menyelesaikan kode program yang diminta pada soal. Hore! Sebelum mengumpulkan pekerjaannya ke *server* Aren, Moo tiba-tiba sakit perut. Ia meninggalkan komputer begitu saja dan pergi ke toilet. Ketika kembali dari toilet, Moo mendapati kode program buatannya sudah diacak-acak oleh Lusi si Usil. Lusi berhasil meloloskan diri tanpa tertangkap. Moo menjadi sedih dan hampir putus asa. Sebagai temannya, Anda tergerak untuk membantunya. Bantulah Moo!

Deskripsi Soal

WS 4 SDA ini mengharuskan Moo untuk membuat sebuah program pohon kalkulator sederhana. Kalkulator ini harus dapat menghitung hasil dan menampilkan sebuah gambar pohon biner sederhana dari setiap ekspresi matematika yang dimasukkan. Setiap ekspresi matematika yang nantinya dimasukkan ke dalam kalkulator dijamin merupakan ekspresi matematika yang valid. Sebuah ekspresi matematika yang valid didefinisikan sebagai berikut.

- 1) Jika x merupakan sebuah bilangan bulat nonnegatif, maka " x " (untuk bagian ini dan selanjutnya tanda kutip hanya untuk kejelasan) merupakan ekspresi matematika valid.
- 2) Jika A merupakan ekspresi matematika yang valid, maka " (A) " juga merupakan ekspresi matematika yang valid.
- 3) Jika A dan B merupakan ekspresi matematika yang valid, maka " $(A+B)$ ", " $(A-B)$ ", " $(A*B)$ ", dan " (A/B) " juga merupakan ekspresi matematika yang valid.
 - Tanda '+' menyatakan ekspresi kiri ditambahkan dengan ekspresi kanan.
 - Tanda '-' menyatakan ekspresi kiri dikurangi dengan ekspresi kanan.
 - Tanda '*' menyatakan ekspresi kiri dikalikan dengan ekspresi kanan.
 - Tanda '/' menyatakan ekspresi kiri dibagi dengan ekspresi kanan. Hasil pembagian selalu dibulatkan ke bawah sehingga tidak ada angka di belakang titik desimal. Namun demikian, bukan berarti bahwa cara pembulatannya dengan langsung menghilangkan angka-angka di belakang titik desimal. Salah satu contoh sebuah ekspresi yang melibatkan pembagian adalah " $((3/2)+5)$ ", hasilnya 6 karena hasil pembagian 3 dengan 2 dibulatkan ke bawah menjadi 1.

Selain menghitung hasilnya, Moo juga diminta untuk mencetak bentuk pohon biner dari setiap ekspresi. Salah satu ilustrasi pohon adalah sebagai berikut.



Gambar 1. Ilustrasi Pohon Biner

Setiap *node* yang ada di dalam pohon biner direpresentasikan dengan sebuah bilangan bulat nonnegatif atau sebuah tanda operasi matematika (tanda kurung tidak perlu direpresentasikan). Bilangan selalu berada di daun, sedangkan tanda operasi selalu berada di *internal node*. *Node-node* dalam pohon biner diletakkan sesuai dengan urutan perhitungan setiap operasi yang ada dalam sebuah ekspresi matematika (ekspresi kiri diletakkan pada anak kiri dan ekspresi kanan diletakkan pada anak kanan) sehingga representasi pohon biner selalu unik. Representasi pohon biner dalam bentuk tulisan mensyaratkan anak kiri diletakkan di bawah anak kanan (lihat contoh).

Diberikan sebuah *template* kode sumber untuk WS 4 ini (bisa diakses dari <http://fpc.cs.ui.ac.id/files/SDA11104.java>), termasuk di dalamnya adalah implementasi representasi pohon biner yang sudah jadi. Dengan demikian, yang perlu Anda buat hanyalah program untuk memproses masukan dan memproses masukan tersebut hingga menjadi pohon biner, serta memanfaatkan pohon biner tersebut agar dapat menghitung hasil akhir dari ekspresi matematika yang diberikan. Buatlah program yang dimaksud untuk membantu Moo!

Format Masukan

Masukan dibaca dari masukan standar. Masukan terdiri dari N buah baris ($N < 50$). Tiap baris berisi sebuah ekspresi matematika valid sesuai definisinya dalam deskripsi soal. Banyak karakter untuk setiap ekspresi minimal 1 dan maksimal 150. Setiap bilangan yang ada di dalam ekspresi dijamin merupakan bilangan bulat nonnegatif yang tidak lebih dari 100.

Format Keluaran

Keluaran ditulis ke keluaran standar. Keluaran terdiri dari N buah bagian yang urutannya sesuai dengan urutan pada masukan. Jika terdapat beberapa bagian, bagian yang satu

langsung diikuti bagian lainnya. Untuk setiap bagian, keluarkan $(M + 2)$ buah baris dengan format berikut.

- Baris pertama berisi ekspresi matematika yang tertulis apa adanya dari masukan, diikuti dengan " = ", lalu diikuti hasil dari ekspresi matematika pada masukan. Hasil tersebut dapat berupa sebuah bilangan bulat atau tulisan "Error" jika terdapat pembagian dengan 0 (atau hasil perhitungan yang menghasilkan 0) dalam ekspresi itu.
- M buah baris berikutnya berisi representasi pohon biner yang dibentuk.
- Baris terakhir merupakan baris kosong.

Contoh Masukan

```
(1+5)
(100/((10*5)-25))
((5-100)/0)
```

Contoh Keluaran

```
(1+5) = 6
[+]
|
#=>5
|
#=>1

(100/((10*5)-25)) = 4
[/]
|
#=>[-]
|   |
|   #=>25
|   |
|   #=>[*]
|   |
|   #=>5
|   |
|   #=>10
|
#=>100

((5-100)/0) = Error
[/]
|
#=>0
|
#=>[-]
|
|
#=>100
|
#=>5
```

Batasan

Kelas-kelas yang sudah disediakan Java yang boleh digunakan hanyalah kelas-kelas dari *package* `java.lang`, `java.io`, dan sebuah *package* lain pilihan Anda. Kelas-kelas selain dari ketiga *package* itu 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.

- Hanya menambahkan kode Anda pada bagian yang ditandai `/*LENGKAPI!*/` dalam *template* kode sumber sesuai dengan petunjuk dalam bentuk komentar pada satu baris sebelumnya. Penambahan, pengubahan, dan/atau penghapusan pada bagian lain akan menghilangkan nilai komponen ini. Penambahan kode yang tidak sesuai petunjuk juga akan menghilangkan nilai komponen ini.
- Terdapat implementasi rekursif yang benar untuk memproses ekspresi matematika ke dalam pohon biner.
- Terdapat implementasi rekursif yang benar untuk menentukan apakah terdapat pembagian dengan 0 dalam sebuah ekspresi matematika.
- Terdapat implementasi rekursif yang benar untuk memproses pohon biner sehingga bisa didapatkan hasil akhir ekspresi matematika.

Jika terdapat kelas-kelas yang tidak boleh digunakan dalam kode sumber berdasarkan batasan di atas, total nilai WS 4 adalah 0.

Lanjutan Cerita

Walaupun Anda berhasil menyelesaikan kodenya, Moo tidak ingin menyalin kode Anda, ia men-*download* kembali *template* WS 4, belajar bersabar dan berusaha menyelesaikan kembali pekerjaannya. Ia hanya ingin mencocokkan keluaran programnya nanti dengan keluaran program Anda berdasarkan masukan yang sama. Walaupun ia seorang pemalu, Moo tidak ingin membuat dirinya sendiri menjadi orang yang memalukan dengan menyalin baik sedikit, sebagian, maupun seluruh kode orang lain yang seharusnya dibuat sendiri. Sekarang, ia pun mulai mengerjakan WS 4. "I'll try my best," katanya.