

# Pemakaian Bandwidth

## Deskripsi

Pak Asep adalah pengelola IT dari suatu perusahaan. Sistem IT yang dikelolanya mencatat pemakaian total bandwidth yang digunakan setiap harinya. Pencatatan dilakukan pada file log setiap harinya jam 23:59:59, dengan menyimpan tanggal dan besarnya gigabyte (GB) pemakaian bandwidth yang terjadi dari jam 00:00:00 sampai 23:59:59 hari tersebut itu.

Suatu hari ia ditanyai pimpinan “jika dihitung mulai hari ke  $d$ , pada hari ke berapakah akumulasi pemakaian mencapai  $m$  GB.” (Catatan: hari ke  $d$  ikut dihitung).

Contoh: Diberikan data pemakaian pemakaiannya sebagaimana pada tabel berikut.

Hari ke	1	2	3	4	5	6	7	8	9
Pemakaian (GB)	5	9	11	9	5	2	2	12	10

Jika dihitung mulai hari ke 3, pada hari ke berapakah terjadi akumulasi pemakaian mencapai 30 GB? Hari ke 3, 11 GB, hari ke 4, menjadi 20 GB, hari ke 5 menjadi 25 GB, hari ke 6 menjadi 27 GB, hari ke 7 menjadi 29 GB, dan hari ke 8 menjadi 41 GB. Jadi akumulasi pemakaian 30 GB terjadi pada hari ke 8.

Banyaknya hari yang tercatat dalam data maksimum ada  $\leq 1.000.000$ , dan untuk data log yang sama pimpinan mengajukan sejumlah pertanyaan demikian sekaligus ( $\leq 10.000$  pertanyaan). Dalam tugas ini anda harus melengkapi suatu program templat dengan menambahkan method untuk melakukan pencarian pada tabel data tersebut yang sudah dibaca ke dalam array.

Program templat (program setengah jadi) sudah memiliki fungsi untuk membaca data (jadi anda tidak perlu pusing dengan format input!) dan menyimpan harga-harga akumulasinya ke dalam array berformat long integer (64 bit integer) bernama  $X$ . Indeks array mulai dari 0, dengan  $X[0]$  berisi harga 0, kemudian  $X[1]$  adalah pemakaian hingga hari 1,  $X[2]$  adalah pemakaian hingga hari ke 2, ...dan seterusnya hingga  $X[N]$  adalah pemakaian hingga hari ke  $N$ . Jadi panjang array adalah  $N+1$ . Dijamin juga data dalam  $X$  yang terbaca akan bersifat nondecreasing (deret menaik atau sama).

Kemudian, ada  $M$  pertanyaan yang harus dijawab. Harga  $M$  dibaca dari file data dan pada setiap baris data berikutnya berisi data setiap pertanyaan yang berupa dua bilangan  $t$  dan  $V$ .

Pada setiap pertanyaan ( $t, V$ ) program harus me-return **harga k terkecil** (program menjawab:  $k$ ) dari semua kemungkinan hari ke  $k$  yang memenuhi  $X[k-1] < (V+X[t]) \leq X[k]$ . Dan, jika ternyata dalam data  $V+X[t] > X[N]$  maka program menjawab: **TIDAK ADA**.

## Contoh Input

```
10
10
10
10
10
```

```
10
10
10
10
10
10
10
1 5
2 15
3 5
4 15
5 0
6 10
7 10
8 5
9 5
10 15
```

### Contoh Output

```
2
4
4
6
5
7
8
9
10
TIDAK ADA
```

### Program Template

Nama file program template adalah **Searching.java** dengan Searching.class sebagai main class di dalamnya. Dalam program sudah ada perintah untuk membaca data log maupun data pertanyaan. Data pertanyaan dibaca dalam iterasi dan setiap pembacaan data pertanyaan program akan memanggil method **find(int t, int V)**. Dalam program method ini masih kosong dan anda harus menuliskan algoritma pencarian datanya (boleh menambahkan method lain dalam rangka penulisan ini) dan menuliskan hasil pencariannya sesuai petunjuk di atas.

Anda dapat mendownload program template ini dengan alamat di:

<http://aren.cs.ui.ac.id/sda/download/Searching.java>

### Batas Waktu

Waktu eksekusi dibatasi (spesifikasi waktu menyusul) sehingga anda perlu mengimplementasikan algoritma Searching yang efisien. Dengan algoritma sequential search memungkinkan untuk data yang kecil tapi untuk kemungkinan data yang besar program anda akan melewati batas waktu ini. Jadi disarankan melakukan binary searching (dengan sejumlah modifikasi, karena yang dicari interval!) pada array X.

## Petunjuk Untuk Submission

Setelah anda melengkapinya dan anda sudah merasa program anda akan berjalan dengan baik (menguji di komputer anda dengan data sesuai spesifikasi di atas), anda dapat mensubmitnya ke server <http://aren.cs.ui.ac.id/sda> dengan account (userid, password) yang akan segera dibagikan.

Setelah login anda akan menemukan menu (dan anda ikuti petunjuk berikut):

### Main Menu

- [Soal-soal Pemrograman](#)
- [Kumpulkan Tugas Pemrograman](#)
- [Meminta Untuk Grading Program](#)

Klik pada menu “[Kumpulkan Tugas pemrograman](#)” dan anda akan menemukan informasi:

### Pengumpulan Tugas Pemrograman

- [Searching](#) [[Browse deskripsi soal](#)]
  - Batas waktu pengumpulan: 23:59:00 14-09-2018
  - Kesempatan pengumpulan tersisa: 20 dari 20
  - Batas waktu eksekusi: 1000 milliseconds
  - Batas memory: 20000000 bytes

Klik pada link [Searching](#) dan anda akan menemukan halaman berisi:

### Sessi Pengumpulan Tugas

\*\* Judul Tugas: Searching  
\*\* File untuk dikumpulkan: Searching.java  
\*\* Batas waktu pengumpulan: 23:59, 14 September 2018  
\*\* Waktu kini: 06:12, 09 September 2018  
\*\* Batas pengumpulan: 20 kali  
\*\* Pengumpulan kini: 0 kali  
\*\* ID>Nama anda: SDAR18002 ()

>>> Ingat:

- Ketikkan nama file dan pathnya (atau bisa mencarinya dengan menekan tombol Browse, memilihnya) lalu menekan tombol Kirim.
- Anda hanya mengumpulkan file dengan nama yang sesuai seperti yang telah ditentukan tersebut di atas.
- File berukuran > 1 MegaByte tidak akan diterima (perlu ijin khusus dari webadmin).
- Jika anda mensubmit melalui batas waktu di atas maka server secara otomatis akan menolaknya (anda harus menyadari adanya waktu yang diperlukan dalam pengiriman file tsb dari PC anda melalui internet ke server).

Choose File

Kirim

Dengan mengklik **Choose File** maka anda akan memilih file untuk disubmit dari direktori lokal. Nama file harus sama dengan nama yang diminta di atas (dalam hal ini Searching.java). Kemudian klik tombol **Kirim** agar file tersebut terkirimkan ke server.

Selanjutnya server akan melakukan Validasi Java dan Kompilasi. Jika tidak ada kesalahan syntax maka anda akan laporan menemukan halaman berikut:

### **Pengumpulan Tugas Pemrograman**

>>> Pengumpulan ini adalah ke 9 dari maksimum 20

#### **Submission Report**

>>> Pengumpulan diterima.

filesize: 6543 bytes

#### **Laporan Validasi Java**

#### **Laporan Kompilasi Java**

#### **Laporan Hasil Uji Coba**

>>> Program anda belum di test run. Anda dapat melakukannya melalui menu "Meminta untuk Grading Program" (jika disediakan)

Untuk meminta server menjalankan program anda anda harus memerintahkan dengan mengklik menu Meminta untuk GradingProgram (dari Menu Utama). Jika program memang tidak ditemukan kesalahan syntax (berhasil di-compile) maka akan muncul halaman:

### **Perform grading to which task?**

- [Searching](#)

Kliklah nama Tugas tersebut (dalam hal ini Searching) maka server akan menjalankannya, kemudian mengujinya dengan data test yang telah disediakan dan melaporkan hasilnya seperti contoh berikut:

### **Grading for MrBlack on Searching**

\*\* Grading server: ---

\*\* Running mode: grading testcase

\*\* Language: java

#### **Running with testcase#1**

Testcase id: SDA18T04609.in

Result: success

Running time : 140 milliseconds

#### **Running with testcase#2**

Testcase id: SDA18T0f99d.in

Result: success

Running time : 790 milliseconds

#### **Summary**

Right answers: 2

Wrong answers: 0

Score: 100

Contoh di atas adalah untuk program yang berhasil memproses seluruh data test dengan benar (sesuai output yang diharapkan) sehingga mendapatkan Score 100. Jika masih belum seluruh benar maka pada bagian Result akan disebutkan "Fail" (dengan informasi kesalahannya).