

IKI20100 2007/2008 (Gasal): Struktur Data & Algoritma  
Fakultas Ilmu Komputer, Universitas Indonesia  
Worksheet: Minggu 2

Pada tutorial minggu ini, anda akan melakukan eksperimen dengan beberapa implementasi algoritma yang dapat menghitung *maximum contiguous subsequence sum* seperti dibahas pada kuliah.

1. Jalankan IDE Eclipse – kalau konfigurasi *workspace*-nya masih sama dengan yang dibuat pada minggu sebelumnya, harusnya pada kolom ter kiri (“*Package Explorer*”) terlihat *Project* yang anda buat sebelumnya. Kalau tidak, buatlah sebuah *Project* baru (lihat worksheet minggu kemarin kalau lupa caranya!).

2. Download dan save file berikut ini:

<https://telaga.cs.ui.ac.id/WebKuliah/IKI20100/tutorial/sda02.jar>

Tambahkan file *sda02.jar* ini sebagai sebuah **library** pada *Project* yang anda gunakan. Carilah class *iki20100.MaxSubSum* yang terdapat di dalamnya. Dapatkah anda melihat method apa saja yang disediakan oleh class tersebut? Class ini mendefinisikan 4 buah method **static**, yakni *maxSubSum1(int[] a)* hingga *maxSubSum4(int[] a)*, yang merupakan implementasi algoritma dengan kompleksitas  $O(N^3)$ ,  $O(N^2)$ ,  $O(N\log N)$ , dan  $O(N)$ .

3. Buatlah sebuah class baru, *TestMaxSubSum*, yang berisi sebuah method **public static void main(String[] args)**. Pada method ini, cobalah membuat sebuah array integer (*int[]*), dan panggil *MaxSubSum.maxSubSum1* sampai *MaxSubSum.maxSubSum4* dengan memberikan array yang anda buat sebagai parameter. Apakah nilai yang dikembalikan memang benar adalah *maximum contiguous subsequence sum*? (**Hint**: cobalah contoh yang diberikan pada slide kuliah. Coba juga array di mana semua nilainya negatif)

4. Pada *TestMaxSubSum*, buatlah sebuah method **static void isiAcak(int[] array, int max)**. Method ini diberikan sebuah array *int[]* yang sudah di-instantiate sebelumnya, dan mengisi setiap elemennya dengan sebuah nilai *int* yang dipilih secara acak antara *-max* sampai *max*. (**Hint**: untuk menghasilkan sebuah *int* secara acak antara 0 sampai *n*, lihatlah method *nextInt(int)* pada class *java.util.Random*. Cobalah lihat dokumentasi API Java yang tersedia di

<https://telaga.cs.ui.ac.id/WebKuliah/IKI20100/docs/api>

untuk keterangan lebih lanjut)

5. Sekarang, coba panggil *MaxSubSum.maxSubSum1* sampai *MaxSubSum.maxSubSum4* dengan argumen berupa array *int[]* yang anda buat secara acak menggunakan method *isiAcak*.

6. Cobalah mengukur dan membandingkan waktu yang dibutuhkan oleh method-method ini untuk memproses array yang sama. Anda dapat menggunakan method *System.currentTimeMillis()* yang mengembalikan sebuah nilai yang menyatakan waktu ketika ia dipanggil (sekali lagi, coba lihat API Java untuk keterangan lebih lanjut). Contohnya seperti demikian:

```
long awal = System.currentTimeMillis();
// lakukan sesuatu di sini...
long akhir = System.currentTimeMillis();
System.out.println("Proses membutuhkan "+(akhir-awal)+" milidetik");
```

7. Cobalah bereksperimen dengan ukuran array yang diberikan sebagai input terhadap method *maxSubSum1* sampai *maxSubSum4*. Kira-kira berapakah ukuran data terbesar yang dapat diproses masing-masing method di bawah waktu 10 detik? Amati hasil eksperimentasi yang anda peroleh – apakah sesuai dengan estimasi laju pertumbuhan  $O(N^3)$ ,  $O(N^2)$ ,  $O(N\log N)$ , dan  $O(N)$ ?

*Selamat mencoba!*