Technische Universität München
Institut für Informatik
Prof. Dr. Ernst W. Mayr

SS 1998
Aufgabenblatt 8
6. Juli 1998

Praktikum Algorithmen-Entwurf

(Abgabetermin: Montag, den 13. Juli 1998, 24.^{00}Uhr)

Longest Increasing Subsequence

Aufgabe 1 Einfache Zahlensequenzen

Implementieren Sie einen Algorithmus, der eine Sequenz $x_1,\ldots,x_n$ paarweise verschiedener, natürlicher Zahlen aus einer Eingabedatei liest und in Zeit $O(n \log n)$ eine längste aufsteigende Teilsequenz berechnet.
Die Eingabedatei enthält in der ersten Zeile die Zahl n und dann in n weiteren Zeilen die Zahlen $x_1,\ldots,x_n$ (eine Zahl pro Zeile). Der Name der Eingabedatei soll Ihrem Programm als Aufrufparameter übergeben werden.
Die Ausgabe Ihres Programms soll aus genau einer Zahl ℓ, gefolgt von '`\n`', bestehen, wobei ℓ die Länge einer längsten aufsteigenden Teilsequenz von $x_1,\ldots,x_n$ ist. Falls Ihr Programm mit Option "`-v`" aufgerufen wird, sollen zusätzlich in ℓ weiteren Zeilen Werte $x_{i_1},\ldots,x_{i_\ell}$ ausgegeben werden, die eine solche längste aufsteigende Teilsequenz darstellen. Alle Ausgaben sollen nach `stdout` bzw. `cout` erfolgen.

Aufgabe 2 Sequenzen von Zahlenpaaren

Gegeben sei eine Sequenz $(x_1, y_1), \ldots, (x_n, y_n)$ von Paaren natürlicher Zahlen. Für alle $1 \leq i < j \leq n$ gelte $x_i \neq x_j$ und $y_i \neq y_j$. Entwickeln Sie einen möglichst effizienten Algorithmus, der eine längste aufsteigende Teilsequenz berechnet, d.h. eine möglichst lange Teilsequenz $(x_{i_1}, y_{i_1}), \ldots, (x_{i_\ell}, y_{i_\ell})$ mit $i_j < i_{j+1}$, $x_{i_j} < x_{i_{j+1}}$ und $y_{i_j} < y_{i_{j+1}}$ für $1 \leq j \leq \ell - 1$.
Die Eingabe soll ähnlich zu Aufgabe 1 aus einer Datei bestehen, die in der ersten Zeile die Zahl n enthält und dann in jeder der n weiteren Zeilen genau zwei durch ein Leerzeichen getrennte Zahlen, die zusammen jeweils ein Paar (x_i, y_i) darstellen.
Die Ausgabe soll wie bei Aufgabe 1 aus genau einer Zahl ℓ, gefolgt von '`\n`', auf `stdout` bzw. `cout` bestehen, wobei ℓ die Länge einer längsten aufsteigenden Teilsequenz ist. Bei Angabe der Option "`-v`" soll Ihr Programm zusätzlich eine längste aufsteigende Teilsequenz ausgeben, wieder mit einem Zahlenpaar pro Zeile. Bitte fügen Sie Ihrer Lösung eine *kurze* Beschreibung Ihres Algorithmus bei.

Hinweise

Beide Programme sind in C oder C++ zu programmieren. Sie sollten Ihre Programme mit Option "-O" übersetzen, um Compiler-Optimierungen einzuschalten. Bei Aufgabe 1 sind nur Standardbibliotheken zugelassen, kein LEDA. Bei Aufgabe 2 dürfen Sie gegebenenfalls LEDA verwenden, z.B. falls Ihnen die Klasse `sortseq` hilfreich erscheint.

– Bitte wenden! –

Weitere Hinweise

Als Eingaben können Sie die Dateien `sequence*.txt` (für Aufgabe 1) und `pair*.txt` (für Aufgabe 2) aus dem Verzeichnis `/usr/local/lea/praktikum/lis` auf den hpmayr-Rechnern verwenden. Um die Korrektheit der Ausgabe Ihrer Programme zu testen, können Sie diese mit der Ausgabe der Programme `lis` (Aufgabe 1) und `lip` (Aufgabe 2) aus `/usr/local/lea/bin` vergleichen. Unten finden Sie zwei kleine Beispiele für korrekte Eingaben und Ausgaben.

Zur Messung der Laufzeit können Sie z.B. das Programm `/bin/time` verwenden. Ihre Programme sollten auch bei den größten Eingaben in vernünftiger Zeit terminieren. Bei Aufgabe 2 müssen Sie daher versuchen, Ideen aus dem in der Praktikumsvorlesung besprochenen LIS-Algorithmus zu verwenden, um eine möglichst effiziente Lösung mit Laufzeit besser als $O(n^2)$ zu erhalten.

Beispiel für Eingabe und Ausgabe zu Aufgabe 1:

Eingabedatei `sequence7.txt`:

```
7
10
19
13
17
14
15
12
```

Beispiellauf von Programm `lis` auf `sequence7.txt`:

```
$ ./lis sequence7.txt
4
$ ./lis -v sequence7.txt
4
10
13
14
15
```

Die gegebene Sequenz $10, 19, 13, 17, 14, 15, 12$ hat als längste aufsteigende Teilsequenz die Sequenz $10, 13, 14, 15$ der Länge 4.

Beispiel für Eingabe und Ausgabe zu Aufgabe 2:

Eingabedatei `pair7.txt`:

```
7
6 3
2 1
4 6
3 4
1 2
7 5
5 7
```

Beispiellauf von Programm `lip` auf `pair7.txt`:

```
$ ./lip pair7.txt
3
$ ./lip -v pair7.txt
3
2 1
3 4
5 7
```

Die Sequenz $(6,3), (2,1), (4,6), (3,4), (1,2), (7,5), (5,7)$ hat als längste aufsteigende Teilsequenz die Sequenz $(2,1), (3,4), (5,7)$ der Länge 3. Ebenso wären $(2,1), (4,6), (5,7)$ oder $(2,1), (3,4), (7,5)$ längste aufsteigende Teilsequenzen.