

## Computer Science COMP3600 in 2002 – Assignment Two

**Due:** 5pm, Friday, Sept. 13  
**Late Penalty:** 25% per day

### Question 1.(35/35)

Given a sequence consisting of  $n$  non-negative integers, solve the following two problems. (i) Find a **strictly decreasing** subsequence which is the longest one among all such subsequences; (ii) Find a **strictly increasing** subsequence to maximize the sum of the elements in the subsequence.

For example, consider a sequence 14, 3, 5, 7, 68, 45, 8, 17, 46, 21, 78, 36. A longest strictly decreasing subsequence of the sequence is 68, 45, 8, and a strictly increasing subsequence with the maximum sum of the elements is 3, 5, 7, 45, 46, 78.

**Your tasks are as follows.**

(a) Devise dynamic programming algorithms for the problems. Requirement: You need to apply the four steps of dynamic programming to this problem and to develop an algorithm for it. Note that no source code is required at this stage. (12/35)

(b) Analyze the time complexities of the two proposed algorithms; (8/35)

(c) Write a program called `decreasing_subseq` for question 1 (i) to find the optimal solutions and print the subsequence. (7/35)

(d) Write a program called `increasing_weight_subseq` for question 1 (ii) to find the optimal solutions and print the subsequence.

(Hint: You may just make minor modifications to the source code for question (i)). (8/35)

**What to submit.**

You can submit the following files and document. Note that you should use the exact names in the following list.

- `makefile`
- `decreasing_subseq.c`
- `increasing_weight_subseq.c`
- `a2_report.ps` (optional)

where the file `makefile` is compulsory. For example, the shell command `make decreasing_subseq` must compile and links to the program `decreasing_subseq`.

You may not need to write a separate report, but your program must contain full documentation. Otherwise, you may lose marks.

## Supplement of Assignment 2

The input of the two algorithms is a file consisting of two lines in which the first line contains the number of elements in the sequence; and the second line is a list of elements separating by space. For example, a file `test 1` contains the following two lines.

```
13
```

```
5 4 7 34 45 17 3 1 37 48 91 23 34
```