CSE530: Algorithms & Complexity
Assignment 1

Antoine Vigneron

March 8, 2018

This homework assignment is due on Monday 19 March, 14:30 at the beginning of the lecture. **Please include your name and student ID.** Each question or subquestion is worth 10 marks. So the total is 130 marks. You should follow the academic integrity rules that are described at the end of the slides of Lecture 0.

---

1. For each pair $f(n), g(n)$ of functions below, which statement is correct: $f(n) = o(g(n))$, or $f(n) = \Omega(g(n))$? You do not need to prove your answer.

$$\begin{array}{lll}
(a) & f(n) = 1000n^2 + 100 & g(n) = n^3 \\
(b) & f(n) = n^2 + 2 & g(n) = 10n^2 - n + 1 \\
(c) & f(n) = 1/\log n & g(n) = \pi \\
(d) & f(n) = n^2 + n\log n & g(n) = n^2 \\
(e) & f(n) = \sqrt{n} & g(n) = n^{1/3}\log n
\end{array}$$

2 In this problem, we consider two functions $f(n)$ and $g(n)$ such that $f(n) = \Theta(g(n))$, and we want to determine whether $f(n) + g(n) = \Theta(f(n) + g(n))$.

(a) Prove that if $f(n) \geqslant 0$ and $g(n) \geqslant 0$ for all n, and $f(n) = \Theta(g(n))$, then $f(n) + g(n) = \Theta(g(n))$.

(b) Give an example of two functions f and g such that $f(n) = \Theta(g(n))$ is true, but $f(n) + g(n) = \Theta(g(n))$ is false.

3. Given an array $A[1 \dots n]$ of n numbers, the 2-MIN problem is to find the two smallest numbers in A. For instance, if $n = 5$ and $A = [4, 3, 7, 2, 8]$ then we should return the pair 2, 3. More formally, we should return the smallest number $A[i^*]$, followed by the smallest number $A[j^*]$ such that $j^* \neq i^*$. We use Algorithm 1 to solve this problem.

(a) Prove that Algorithm 1 is correct using the loop invariant method. (You only need to prove a loop invariant for the outer loop, not for the inner loop.)

(b) Assume that each execution of Line i takes time c_i, where c_i is a constant. Give the running time of Algorithm 1 as a function of n, and using the constants $c_2, c_3, \dots, c_7$.

(c) Give the running time of Algorithm 1 using the $\Theta(\cdot)$ notation, and justify your answer.

Algorithm 1

```
procedure SLOW 2-MIN(A[1...n])
    x ← A[1], y ← A[2]
    for i ← 1, n − 1 do
        for j ← i + 1, n do
            if A[i] + A[j] < x + y then
                x ← A[i]
                y ← A[j]
    return min(x, y), max(x, y)
```

(d) Give an asymptotically faster algorithm for the 2-MIN problem. (You should give its pseudocode and its running time using the $\Theta(.)$ notation.)

4. We consider the Dynamic Time Warping (DTW) problem with outliers. More precisely, given two point sequences $A = (a_1, \dots, a_m)$ and $B = (b_1, \dots, b_n)$, and an integer $p < m$, we consider the problem of computing the smallest possible value of $DTW(A', B)$ where A' is obtained by deleting p points from A. (See Figure 1.) So we need to minimize over all possible choices of these p points of A, and over all possible couplings for each such choice of p points. These p points are called the *outliers*. The motivation could be that A contains p erroneous points that we would like to identify and discard.

(a) Give an $O(mnp^2)$-time algorithm for this problem: Given A, B and p, it should return the minimum value of $DTW(A', B)$ over all possible sequences A' obtained by deleting p points from A.

(b) Show how to modify the algorithm from (a) so that it returns the p outliers efficiently.

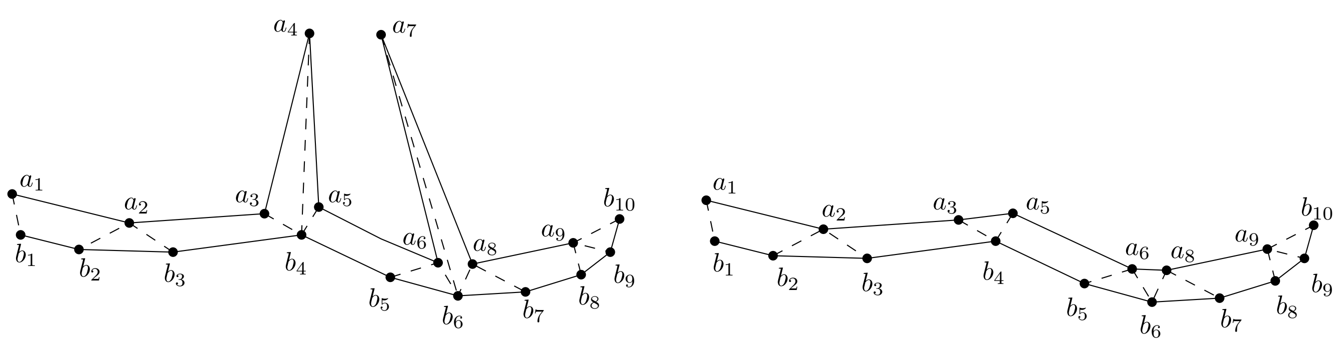

Figure 1: (Left) two point sequences $A = (a_1, \dots, a_9)$ and $b = (b_1, \dots, b_{10})$ and the optimal coupling (dashed) corresponding to $DTW(A, B)$. (Right) If we allow $p = 2$ outliers, then the points a_4 and a_7 are deleted from A, thus obtaining $A' = (a_1, a_2, a_3, a_5, a_6, a_8, a_9)$. The optimal coupling for $DTW(A', B)$ is represented by dashed line segments.