UNIVERSITY OF BIRMINGHAM

School of Computer Science

Algorithms and Complexity (Extended)

Week 12 Continuous Assessment

Algorithms and Complexity (Extended)

Week 12 Continuous Assessment

This assessment has a total of 60 marks. You may want to read all the questions before starting to solve them.

Please note that you can only submit a single pdf file as your solutions.

You can use any results (theorems, algorithms, etc.) covered in the module (lectures and assignments).

Question 1

Let $n \geq 3$. Design a $2^n \cdot n^{O(1)}$ time algorithm for Almost Traveling Salesman Problem (Almost-TSP) problem which is defined as follows:

- Input: A set $C = \{c_1, c_2, \ldots, c_n\}$ of n cities and a distance function $\texttt{dist} : C \times C \to \mathbb{R}^{\geq 0}$. Note that $\texttt{dist}(c, c) = 0$ for each $c \in C$.
- Output: Return the distance of a tour which minimizes the total distance traveled when starting and ending at c_1 while visiting **exactly** $n-2$ cities from C.

Explain both the correctness and running time of your algorithm. **[20 marks]**

Question 2

The function $\mathsf{d} : \{0,1\}^3 \to \{0,1\}$ is given by:

$$\mathsf{d}(x, y, z) = 1 \iff x = y = 1 \text{ or } x = z = 0$$

(a) Write a CNF computing $\mathsf{d}(x, y, z)$. **[5 marks]**

(b) We may build expressions using d and the Boolean constant 0 to compute other Boolean functions. E.g. $d(x, x, x)$ computes the Boolean constant 1, and $d(0, 1, x)$ (which is shorthand for $d(0, d(x, x, x), x)$) computes $\neg x$.

Build expressions using d and 0 computing $x \vee y$ and $x \wedge y$. **[5 marks]**

(c) We may build circuits with non-source nodes labelled by either d or 0, having 3 or 0 incoming edges respectively. Call such circuits $\{\mathsf{d}, 0\}$-circuits.

Using (b), show that any Boolean circuit C (as usual, with non-source nodes labelled $\neg$, $\vee$ or $\wedge$) can be polynomially transformed to a $\{\mathsf{d}, 0\}$-circuit C' computing the same Boolean function, such that $depth(C') = O(depth(C))$. **[5 marks]**

Question 3

Recall that the size of a CNF is its number of literal occurrences. A **minimal** CNF computing a Boolean function f is one of smallest size.

Write $\vec{x} = (x_1, \ldots, x_n)$ and $\vec{y} = (y_1, \ldots, y_n)$ and $\vec{z} = (z_1, \ldots, z_n)$. Consider the Boolean function:

$$f(\vec{x}, \vec{y}, \vec{z}) \quad := \quad \bigvee_{i=1}^{n} (x_i \wedge y_i \wedge z_i)$$

Let $\bigwedge_{j=1}^{m} \bigvee L_j$ be a **minimal** (i.e. of smallest size) CNF computing $f(\vec{x}, \vec{y}, \vec{z})$, where each L_j is a set of literals over $\vec{x}, \vec{y}, \vec{z}$.

(a) Show that, for fixed Boolean inputs, if $\bigvee L_j = 0$ for some $j = 1, \ldots, m$, then $f(\vec{x}, \vec{y}, \vec{z}) = 0$. **[5 marks]**

(b) Show that, for fixed Boolean inputs, if $f(\vec{x}, \vec{y}, \vec{z}) = 0$ then, for all $i = 1, \ldots, n$, one (or more) of x_i, y_i, z_i is 0. **[5 marks]**

Question 4

Read the document available at https://canvas.bham.ac.uk/courses/46223/files/10827563. Summarize (in roughly one page) in your own words what you understand about the Exponential Time Hypothesis (ETH).

[15 marks]