**MTH6105 – Algorithmic Graph Theory** **Spring 2024**
**Problem Sheet 4** **F. Fischer**

---

You are expected to **attempt all exercises** before the seminar and to **actively participate** in the seminar itself.

---

1. (a) For functions $f, g, h : \mathbb{N} \to \mathbb{R}^+$ show the following:
   (i) $f(n) + g(n)$ is $O(\max\{f(n), g(n)\})$.
   (ii) if $f(n)$ is $O(g(n))$ and $g(n)$ is $O(h(n))$, then $f(n)$ is $O(h(n))$.

   (b) Let $f(n) = a_0 + a_1 n + a_2 n^2 + a_3 n^3$, where $a_0, a_1, a_2, a_3 \in \mathbb{Z}$ are constants. Show that $f(n)$ is $O(n^3)$.

   (c) Show that $2^{2n}$ is *not* $O(2^n)$. To this end, you may want to assume that it was and derive a contradiction.

   **Note:** This type of question looks intimidating, but really it is all bark and no bite. Recalling the relevant definition, all we need to do to show that $f(n)$ is $O(g(n))$ is find constants $c$ and $n_0$ with certain properties and verify that they indeed satisfy these properties.

2. Euclid's algorithm determines the greatest common divisor $\gcd(a, b)$ of two non-negative integers $a \geq b$ by setting $r_0 = a$, $r_1 = b$, and then repeating the following steps for rounds $n = 2, 3, 4, \ldots$:

   - If $r_{n-1} = 0$ then stop, output $\gcd(a, b) = r_{n-2}$.
   - Find $q_n$ and $r_n$ such that $r_{n-2} = q_n r_{n-1} + r_n$ and $0 \leq r_n < r_{n-1}$.

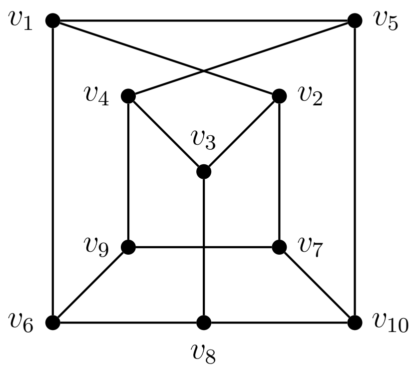
   Note that $(r_n)_{n \geq 0}$ is a decreasing sequence of non-negative integers and the algorithm thus stops after a finite number of rounds. Note further that it makes sense to say that the size of the input of the algorithm is $\log_2 a + \log_2 b$, because this is the number of digits of $a$ and $b$ as binary numbers.

   (a) If $r_n$ were to decrease by 1 in each round, how many rounds would the algorithm run for?

   (b) Show that $r_n$ in fact decreases significantly over two consecutive rounds of the algorithm, namely $r_n < r_{n-2}/2$. You may want to distinguish among the three cases where $r_{n-1} = r_{n-2}$, $r_{n-2}/2 < r_{n-1} < r_{n-2}$, or $r_{n-1} \leq r_{n-2}/2$.

   (c) Give an upper bound on the maximum number of rounds of the algorithm in terms of the size of the input to the problem. You may want to argue that after a certain number $k$ of rounds, $r_k < 1$ and the algorithm must therefore have stopped.

3. Consider the following graph.




(a) Apply breadth-first search to the graph, starting from $v_1$. Give the tree $T$ after each iteration of the algorithm.

(b) Apply depth-first search to the graph, starting from $v_1$. Give the tree $T$ after each iteration of the algorithm.

(c) Which of the algorithms can be used to find a spanning tree of the graph? Draw such a spanning tree.

(d) Which of the algorithms can be used to find a shortest $v_1$–$v_8$-path in the graph? Give such a path.