The aim of this assignment is to give you some practice with various flavours of induction. For each question below you will present a proof by induction. For full marks you need to make it clear to the reader that the base case(s) is/are verified, that the inductive step follows for each element of the domain (typically the natural numbers), where the inductive hypothesis is used, and that it is used in a valid case.

Your assignment must be typed to produce a PDF document a1.pdf (hand-written submissions are not acceptable). You may work on the assignment in groups of 1 or 2, and submit a single assignment for the entire group on MarkUs

1. Recall bipartite graphs. Consider the following definitions:

bipartite graph: Undirected graph $G = (V, E)$ is bipartite if and only if there exist $V_1, V_2$ such that $V = V_1 \cup V_2$, $V_1 \cap V_2 = \emptyset$, and every edge in $E$ has one endpoint in $V_1$ and the other in $V_2$.

$P(n)$: Every bipartite graph on $n$ vertices has no more than $n^2/4$ edges.

(a) Assume $P(234)$. Can you use this$^1$ to prove that $P(235)$ follows? Explain why, or why not.

(b) Assume $P(235)$. Can you use this$^2$ to prove that $P(236)$ follows? Explain why or why not.

(c) Use what you’ve learned from the previous two answers to construct a proof by simple induction that: $\forall n \in \mathbb{N}, P(n)$. Note: There are proofs of this claim that are not by simple induction, but those proofs will receive no marks. Hint: You probably need to strengthen the claim in order to devise a successful inductive hypothesis. If this seems mysterious, revisit the previous two answers...

2. Define function $f$ as follows:

$$f(n) = \begin{cases} 3 & \text{if } n = 0 \\ \lfloor f(\lfloor \log_3 n \rfloor) \rfloor^2 + f(\lfloor \log_3 n \rfloor) & \text{if } n > 0 \end{cases}$$

Define predicate $P(n)$: “$f(n)$ is a multiple of 4.”

(a) Assume $P(3)$. Can you use this$^3$ to prove $P(29)$? Explain why or why not.

(b) Assume $P(4)$. Can you use this$^4$ to prove $P(29)$? Explain why or why not.

(c) Use complete induction to prove $\forall n \in \mathbb{N}, n > 0 \Rightarrow P(n)$.

---

$^1$If you say yes, $P(234)$ must be a necessary part of your proof.

$^2$If you say yes, $P(235)$ must be a necessary part of your proof.

$^3$If you say yes, $P(3)$ must be a necessary part of your proof.

$^4$If you say yes, $P(4)$ must be a necessary part of your proof.
3. Use the Principle of Well-Ordering to derive a contradiction that proves there are no positive integers \(x, y, z\) such that:

\[5x^3 + 50y^3 = 3z^3\]

You may assume, without proof, that if a prime number \(p\) divides a perfect cube \(n^3\), then \(p\) also divides \(n\).

4. Define \(T\) as the smallest set of strings that satisfies:

- "**" \(\in T\)
- if \(t_1, t_2 \in T\) then their parenthesized concatenation \((t_1 t_2)\) \(\in T\).

Some examples: "**", "(**)", "(*(**))" are all in \(T\).

Now read over these four Python functions:

```python
def left_count(s: str) -> int:
    ""
    Return the number of "(" in s
    ""
    return s.count("(")

def double_count(s: str) -> int:
    ""
    Return the number of "((" plus number of ")")", including possible overlaps.
    ""
    return (len([s[i:] for i in range(len(s)) if s[i:].startswith("((")]
             + len([s[:i] for i in range(len(s) + 1) if s[:i].endswith("))")]))

def left_surplus(s: str, i: int) -> int:
    ""
    Return the number of "(" minus the number of ")" in s[:i]
    ""
    return s.count("", 0, i) - s.count("", 0, i)

def max_left_surplus(s: str) -> int:
    ""
    Return the maximum left surplus for all prefixes of s.
    ""
    return max([left_surplus(s, i) for i in range(len(s))] + [0])
```

(a) Use structural induction on \(T\) to prove:

\[\forall t \in T, \text{left_count}(t) \leq 2^{\text{max_left_surplus}(t)} - 1\]

(b) Use structural induction on \(T\) to prove: [edit:] error fixed September 9

\[\forall t \in T, \text{double_count}(t) = \begin{cases} 0 & \text{if } t = "*" \\ \text{left_count}(t) - 1 & \text{otherwise} \end{cases}\]