

Question 1: Smart Bubble Sort Decision Tree

Question 2: Program to Check if the “else” Branch is Taken

1. Assume such a program (C) exists
2. Construct Q as follows:

```
def Q():  
    if P(i) != P(i):  
        print('if')  
    else:  
        print('else')
```

Q will execute the else branch if and only if P terminates on i. Therefore, if C can tell whether Q will execute the else branch, C can tell if P terminates on i, and solves the halting problem. However, we know this is impossible, so we have a contradiction – our assumption that C exists must be wrong.

Question 3: Can a sorting algorithm run in \log time on $n!^{(1/\sqrt{n})}$ of its inputs?

Longest path has to take $\log_2(n!)$ comparisons with $n!$ leaves. If we have k leaves, it will take $\log_2(k)$. Here, this is: $\log(n!^{(1/\sqrt{n})})$. Taking the power to the front, you get: $1/\sqrt{n} * \log(n!)$.

We already know $\log(n!)$ [...]

Question 4: Separation

[...]

Question 5: Prefix Code Forbidden Numbers

[...]

Question 6

(a) Modularity Inequality

- many people gave size-based version, when output-based was what was wanted

(b) True

- can show this using Gödel encoding

(c) True

- can show this as prime numbers is a subset of \mathbb{N}

(d) False

- power set of \mathbb{N} is equivalent to set of infinite binary sequences