# ECE345 HW1

Homework Group #85
Yanni Alan Alevras | yanni.alevras@mail.utoronto.ca | 1009330706
Evan Banerjee | evan.banerjee@mail.utoronto.ca | 1009682309
Gautam | gautam.makheja@mail.utoronto.ca | 1008788974

Total Pages: 7

Thursday 19th September, 2024

# 1 Question 1 Examples of this is a second test

a.

**Theorem.** Give a combinatorial argument to prove that

$$\sum_{k=0}^{n} \binom{n}{k} 2^k = 3^n$$

*Proof.* Proof by combinatorial Argument

imagine you have a bucket with three balls in it, where each ball is either red, green, or blue.

let's say each time you pick one of the balls out of the bucket, you make a dash on a piece of paper in whatever color the ball is, and then put the ball back. Assume that you write the dashes in order so that after repeating this process n times, you end up with a sequence of n dashes in order, where each dash is either red, green, or blue.

since each dash was one of three possible choices, there are $3^n$ possible sequences that you can draw with this process.

□

b. Time complexity: $\mathcal{O}(n)$

*Proof.* ($\Leftarrow$) Assume by Contradiction,

($\Rightarrow$) □

## 2 Question 2 Example Pseudocode

```
function GALE-SHAPLEY(E, S)
    initialize all employers in E and students in S to unmatched
    while an unmatched employer with at least one student on its preference list remains do
        choose such an employer e ∈ E
        make offer to next student s ∈ S on e's preference list
        if s is unmatched then
            Match e with s                                          ▷ s accepts e's offer
        else if s prefers e to their current employer e′ then
            Unmatch s and e′                                        ▷ s rejects e′
            Match e with s                                          ▷ s accepts e's offer
        end if
        cross s off e's preference list
    end while
    report the set of matched pairs as the final matching
end function
```

# 3 Question 3

Figure 1: Example image

This figure shows xxx

# 4 Question 4 Induction

a.

**Theorem.** *Some theorem here.*

*Proof.* Proof by induction

**Base Step**: When $n = 1$, is true.

**Induction Hypothesis**: Suppose is true

**Induction Step**: Consider when $\square$

b. Time complexity: $\mathcal{O}(n)$

*Proof.* ($\Leftarrow$) Assume by Contradiction,

($\Rightarrow$) $\square$

# 5 Question 5 Probability

a.

**Theorem.** *Some theorem here.*

*Proof.* Proof by induction

**Base Step**: When $n = 1$, is true.

**Induction Hypothesis**: Suppose is true

**Induction Step**: Consider when $\square$

b. Time complexity: $\mathcal{O}(n)$

*Proof.* ($\Leftarrow$) Assume by Contradiction,

($\Rightarrow$) $\square$

## 6 Question 6 Graphs, Proof by Contradiction

*Proof.* Assume by Contradiction,

Assume $d(u, w) + d(w, v) < d(u, v)$.

This implies that there is a path from $u$ to $v$ via $w$ whose total distance is less than the direct distance between $u$ and $v$.

However, by the definition of $d(u, v)$, it is the shortest distance between $u$ and $v$. Therefore, no other path, including one through $w$, can have a smaller distance.

This contradiction arises from our assumption, so the assumption must be false. Hence, we conclude that $d(u, w) + d(w, v) \geq d(u, v)$. $\square$

# 7 Question 7 Trees, Proof by induction

**Definition:** A *stable parent* is defined as a node with two leaves. Let:

- $n$ be the number of nodes,
- $l$ be the number of leaves,
- $p$ be the number of stable parents.

**Proof.** Proof by induction.

Assume the inductive hypothesis:

$$l_n = p_n + 1 \quad \Rightarrow \quad l_{n+1} = p_{n+1} + 1$$

We will now consider two cases:

**Case 1:** Add a child to a leaf.

- Since the leaf gains a child, while a new leaf is added, we have $l_{n+1} = l_n$.
- Since no node with 1 child gains another child, $p_{n+1} = p_n$.
- Therefore, $l_{n+1} = l_n = p_n + 1 = p_{n+1} + 1$.

**Case 2:** Add a child to a node with 1 child already.

- Since no leaf stops being a leaf, you have simply added a leaf, so $l_{n+1} = l_n + 1$.
- Since the node with 1 child gains another child, a new stable parent is formed, so $p_{n+1} = p_n + 1$.
- Therefore, $l_{n+1} = l_n + 1 = p_{n+1} + 1 = p_n + 1 + 1$.

**Base case:** For $n = 1$, we have:

- 1 node implies 1 leaf and 0 stable parents.
- Thus, $l_1 = 1$ and $p_1 = 0$.
- Therefore, $1 = 0 + 1$, which satisfies $l_1 = p_1 + 1$.

Q.E.D.