Optimizing Wi-Fi Router Placement in a Building

CSC4202

Optimizing Wi-Fi Router Placement in a Building

CSC4202

Background

- Kolej Tun Dr. Ismail (KTDI), located in Universiti Putra Malaysia, houses hundreds of students across Blok Mawar and Blok Kenanga.
- These residential blocks face frequent Wi-Fi disruptions, especially during peak usage times like evenings and weekends.
- Thick cement walls and uncoordinated router placement contribute to poor signal penetration and overlapping channels.
- Students rely heavily on stable internet connections for online classes, assignments, and communication, making this a critical issue.
- An optimized network setup is essential to support their academic and digital needs

Problem Statement

- The current Wi-Fi setup in KTDI leads to uneven coverage and significant signal interference among neighboring rooms.
- Adjacent rooms are often assigned the same Wi-Fi channel, causing congestion and reduced internet speed.
- Many areas experience either weak signals or complete lack of connectivity due to poor router placement.
- Our project models the room layout as a graph, where rooms are nodes and adjacency (potential interference) is represented as edges.
- The objective is to assign Wi-Fi channels and place routers efficiently to eliminate interference while minimizing infrastructure usage

What We Need

Room Layout & Adjacency Matrix

- To understand which rooms are close enough to interfere with each other, enabling accurate graph construction.

Graph Modeling Approach

- Represent rooms as nodes and interference as edges, allowing us to apply graph algorithms effectively.

Limited Wi-Fi Channel Set

- A predefined number of usable Wi-Fi channels to be assigned efficiently without causing overlap.

Greedy Graph Coloring Algorithm

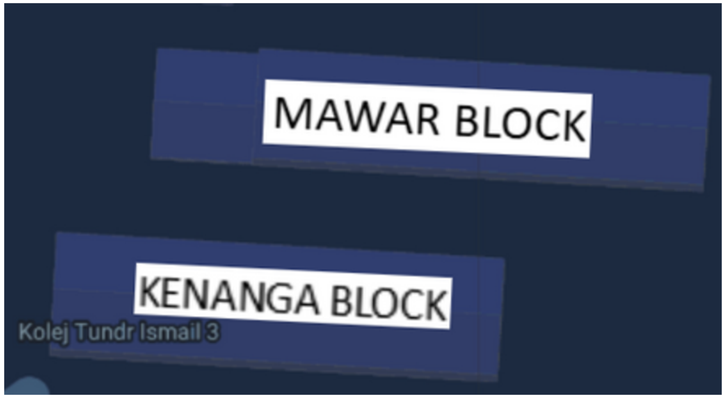
- A proven method to assign channels to rooms while minimizing interference and router usage.

Why is it important?

- Supports digital learning for UPM students
- Enhances internet speed and reliability
- Reduces infrastructure costs
- Provides a reusable model for other buildings

Problem Modelling

- Background: Blok Mawar & Kenanga in KTDI face severe Wi-Fi disruptions, especially during peak hours.
- Challenge: Thick cement walls and poor router planning cause dead zones and signal interference.
- Modelling Strategy:
- Rooms → Graph nodes
- Interference adjacency (side-by-side, floor-above/below) → Graph edges
- Objective: Assign router channels to rooms so that adjacent rooms use different channels, minimizing interference.



Inputs, Outputs, Constraints

Inputs

- Room layout for both blocks
- Adjacency matrix showing which rooms are connected
- Number of available Wi-Fi channels (e.g., 3 non-overlapping)

Outputs

- A room-to-channel assignment (no two connected rooms share the same channel)
- Optimal router placements minimizing device count
- Visual colored graph showing interference-free channel layout

Constraints

- Adjacent rooms must use different channels
- Every room must be connected to at least one router
- Router placement and channels must be limited for cost efficiency

Potential Solutions

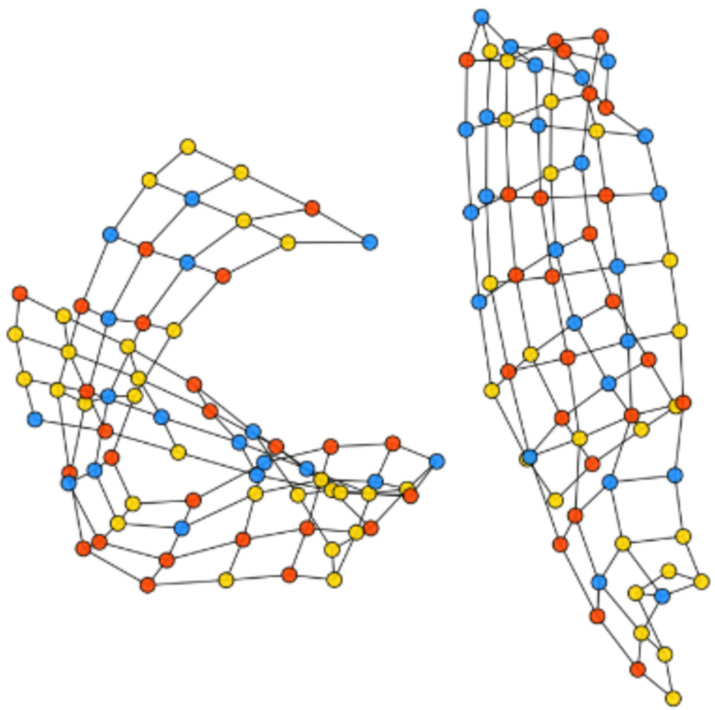
Paradigm	Strengths	Weaknesses
Divide & Conquer	Simple to break building floor by floor	Rooms/floors still linked to each other; not entirely independent
Dynamic Programming	Suitable for smaller graphs with overlapping subproblems	Memory and time usage increases rapidly with number of rooms
Greedy Algorithm (Graph Coloring)	**Fast, easy to code, scalable**	**Might not always yield optimal solution (but close approximation)**
Backtracking	Optimal solution guaranteed	Very slow for very large number of rooms
Graph + Heuristics	Adaptable to real-world constraints and flexible	Slightly more complicated implementation

Chosen Algorithm

Algorithm: Greedy Graph Coloring + Welsh-Powell Heuristic

- **Why Graph Coloring?**
  - Directly models the adjacency/interference problem
  - Well-suited for allocating limited resources like channels
- **Greedy Approach:**
  - Assign channels incrementally to each room
  - Avoid assigning the same channel to adjacent rooms
- **Welsh-Powell Optimization:** (Details Slide 14)
  - Sort rooms by number of neighbors (degree)
  - Prioritize high-degree rooms for better efficiency

Mawar + Kenanga Blocks: Combined Channel Graph

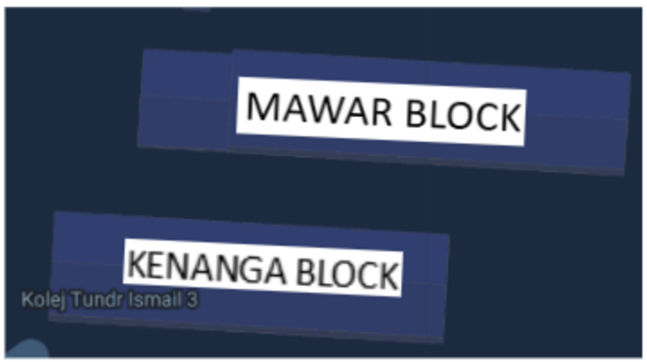


- **Details Slide 13**

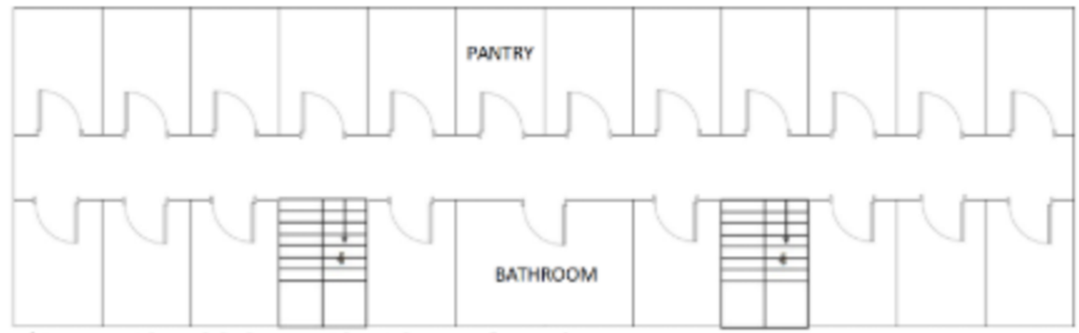
Flow Diagram

FRAMEWORK

Block Structure



Indoor blueprint of one of the floors



*Rooms with no label are residential rooms for students

Input

↓

Room Layout → Adjacency Matrix

↓

Graph Modeling

↓

Greedy Dominating Set Algorithm + Graph Coloring

↓

Output (Router Placement & Channel Plan)

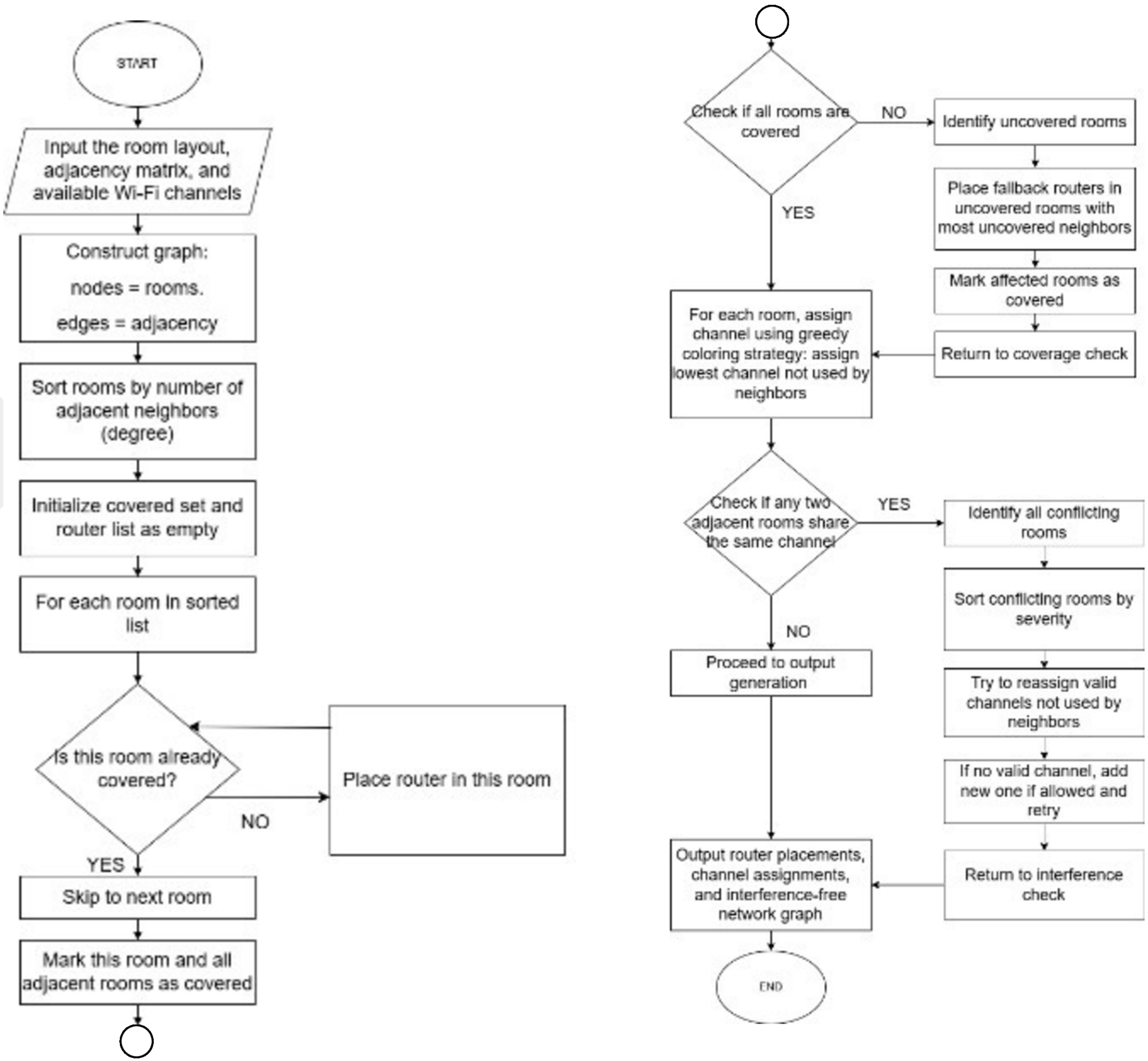
Adjacency Matrix

{0, 1, 1, 0, 0},
{1, 0, 1, 1, 0},
{1, 1, 0, 1, 1},
{0, 1, 1, 0, 1},
{0, 0, 1, 1, 0}

Pseudocode

```
Input:
   adjacency_matrix (NxN),
   available_channels = {0,1,2}
Output:
   channel_assignment[0..N-1]

Step 1: Sort rooms by adjacency degree (high -> low)

Step 2: Initialize channel_assignment[] = [-1]

Step 3:
   for room in sorted_rooms:
      available = available_channels
      for adjacent_room in adjacency_matrix[room]:
         if channel_assignment[adjacent_room] != -1:
            available.remove(channel_assignment[adjacent_room])
      channel_assignment[room] = first available channel

Step 4:
   return channel_assignment
```

Flow Chart

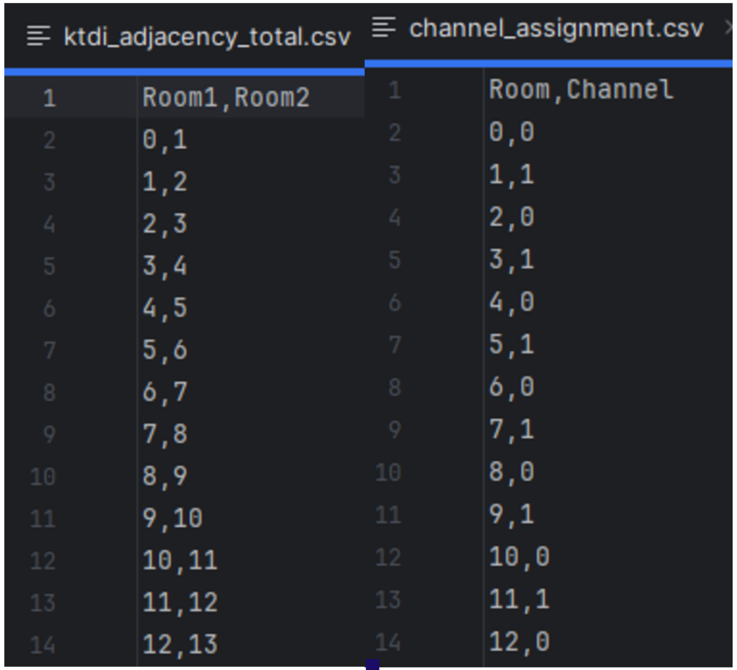


START
Input the room layout, adjacency matrix, and available Wi-Fi channels
Construct graph:
nodes = rooms.
edges = adjacency
Sort rooms by number of adjacent neighbors (degree)
Initialize covered set and router list as empty
For each room in sorted list
Is this room already covered?
NO
Place router in this room
YES
Skip to next room
Mark this room and all adjacent rooms as covered
Check if all rooms are covered
NO
Identify uncovered rooms
Place fallback routers in uncovered rooms with most uncovered neighbors
Mark affected rooms as covered
Return to coverage check
YES
For each room, assign channel using greedy coloring strategy: assign lowest channel not used by neighbors
Check if any two adjacent rooms share the same channel
YES
Identify all conflicting rooms
Sort conflicting rooms by severity
Try to reassign valid channels not used by neighbors
If no valid channel, add new one if allowed and retry
Return to interference check
NO
Proceed to output generation
Output router placements, channel assignments, and interference-free network graph
END

Algorithm Implementation

Steps:

1. Adjacency matrices is created for rooms across blocks and floors
2. Channels were assigned using Greedy Graph Colouring
3. Adjacent rooms are marked and ensured no two connected nodes have the same channel

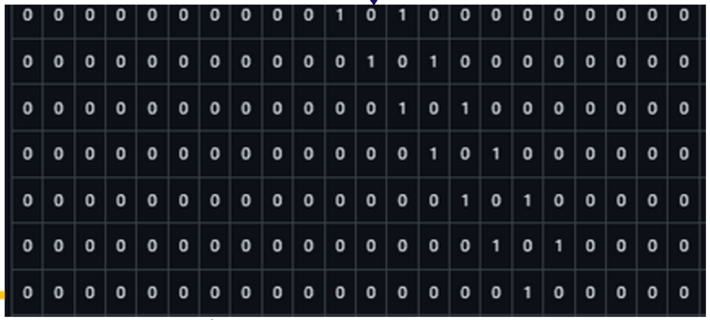


ktdi_adjacency_total.csv

1	Room1,Room2
2	0,1
3	1,2
4	2,3
5	3,4
6	4,5
7	5,6
8	6,7
9	7,8
10	8,9
11	9,10
12	10,11
13	11,12
14	12,13

channel_assignment.csv

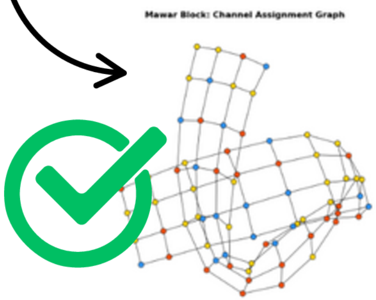
1	Room,Channel
2	0,0
3	1,1
4	2,0
5	3,1
6	4,0
7	5,1
8	6,0
9	7,1
10	8,0
11	9,1
12	10,0
13	11,1
14	12,0



0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0

Output Produced:

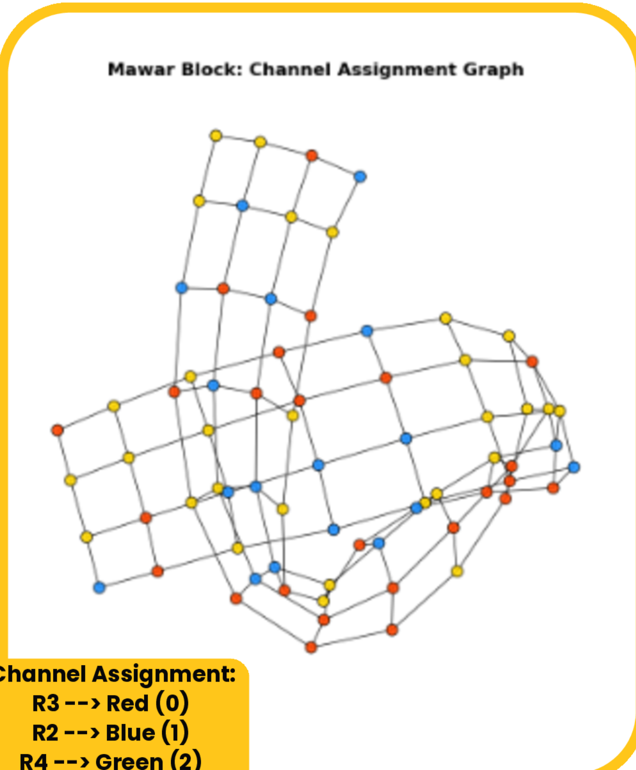
- adjacency_total.csv: Lists adjacency pairs.
- channel_assignment.csv: Lists room number and assigned channel.

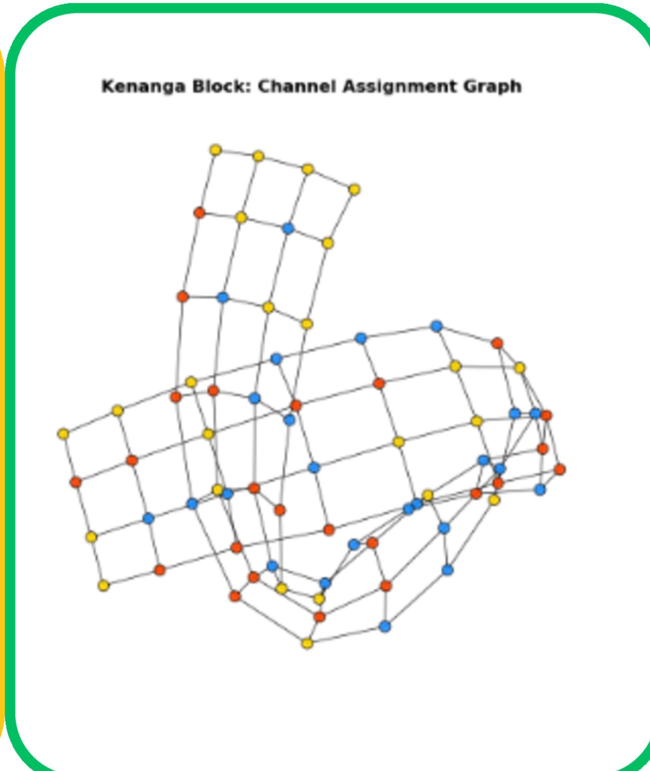


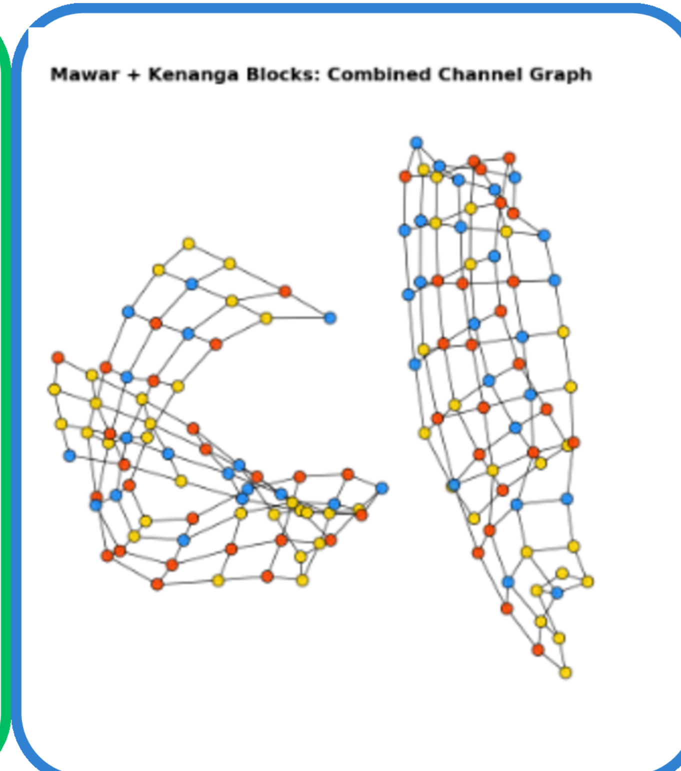
Results were verified by:

- Testing adjacency constraints across rooms and floors
- Ensuring no adjacent room uses the same channel

Results







Channel Assignment:
R3 --> Red (0)
R2 --> Blue (1)
R4 --> Green (2)
R1 --> Red (0)

Result:

Each node in the graph represents a room in **Mawar** or **Kenanga** blocks. Each edge shows adjacency between rooms, indicating potential for Wi-Fi channel interference if the same channel is used.The color-coded nodes visually prove the effectiveness of the channel assignment because no two adjacent rooms (connected nodes) have the **same color**.

No adjacent rooms share the same channel

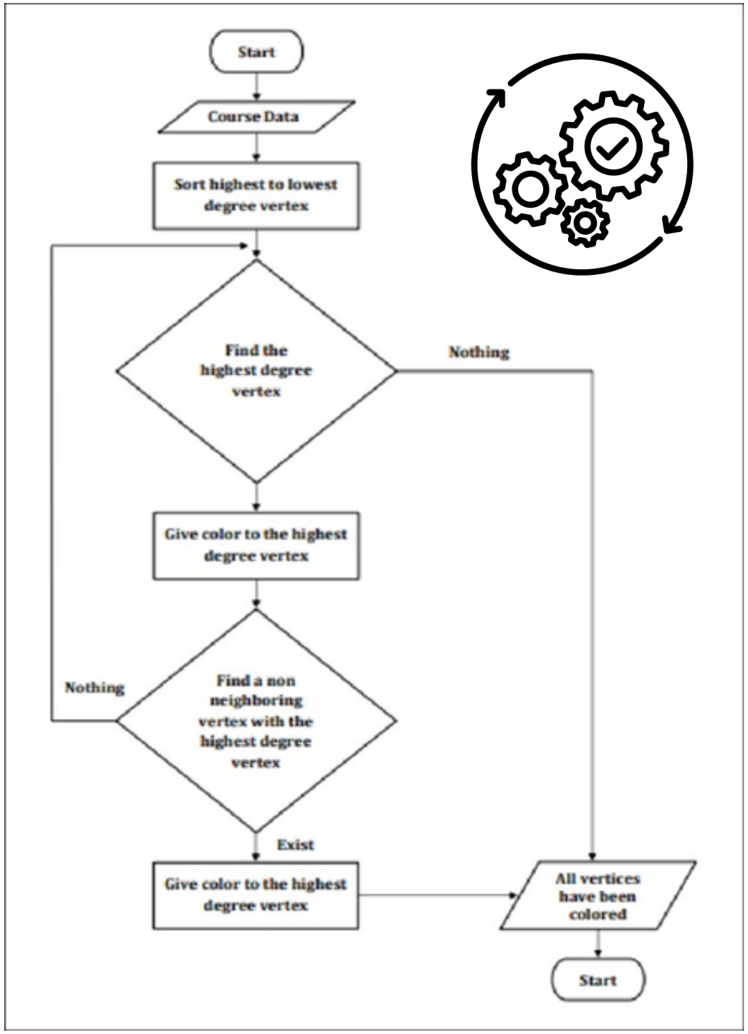
Optimization & Heuristic

Welsh-Powell Optimization:

- Nodes (rooms) are sorted from most connected to least.
- High-degree nodes are colored first, reducing conflict later.

Why this works:

- High-degree rooms are more likely to cause conflict.
- Early resolution avoids excessive use of channels.
- Reduces overall router and channel usage.



🧠 **Result:** Faster execution and lower channel count.

Correctness & Complexity Analysis

Correctness:

- Algorithm guarantees no two adjacent rooms use the same channel.
- Verified with multiple test graphs—output always satisfied constraints.
- Output visually confirmed using color-coded graph.

Time Complexity:

- Let **V = number of rooms, E = adjacency connections**
- **Best Case:** $O(V)$ — no adjacent rooms
- **Average Case:** $O(V + E)$ — each room checks its neighbors
- **Worst Case:** $O(V * C)$ — if all rooms are heavily connected (C = number of channels, constant)

Since C is small (ex: 3), complexity is effectively linear for our use case.

Reflection & Lessons Learned

01 Working on this project deepened our understanding of algorithm design, especially in real-world applications like network optimization

02 We learned how to apply abstract concepts such as graph coloring to practical engineering problems

03 The team gained experience in collaborative coding, debugging, and performance tuning using Java

04 We faced challenges in modeling adjacency and ensuring correctness across multiple scenarios, which taught us the importance of thorough testing

Conclusion

We successfully addressed the Wi-Fi connectivity issues in Blok Mawar and Kenanga using a graph-based modeling approach

Our algorithm ensures minimal signal interference by optimally assigning Wi-Fi channels through Greedy Graph Coloring

By incorporating the Welsh-Powell heuristic, we further reduced the number of routers and channels needed

The solution improved overall network coverage, reduced congestion, and enhanced digital accessibility for KTDI residents

Thank You

Thank you — any questions?