PRACTICAL FILE

MODELING AND SIMULATION LAB

(CS 603)

BE CSE 6TH SEM

(GROUP-4)

University Institute of Engineering and Technology (UIET), Panjab University, Chandigarh, India- 160014

Under the guidance of

Priyanka Mam

Department of Computer Science and Engineering

Submitted By

Ojas Arora

Roll No: UE223073

Practical 6

Aim

Simulation of Fire Spread in a Forest using cellular automata.

Introduction to Fire Spread in a Forest Simulation

Forest fires are a natural phenomenon that can cause widespread destruction to ecosystems. To study and predict the behavior of fire spread, **cellular automata (CA)** can be used as a computational model. **Cellular automata** are grid-based systems where each cell follows simple rules based on its neighbors, leading to complex emergent behavior.

Concept of Cellular Automata in Fire Spread

A **cellular automaton** consists of:

- A **grid** where each cell represents a portion of the forest.
- A set of **states** (e.g., Tree, Burning, Empty).
- **Rules** determining how fire spreads between neighboring cells.

Each tree can be in one of the following states:

- **Tree (Green):** Unburned and susceptible to catching fire.
- **Burning (Red):** Currently on fire and will spread flames.
- **Empty (Gray):** Already burned down or cleared.

At its core, this simulation helps in understanding **how fire propagates through different terrains** and the **factors that influence its speed and intensity**. The **spread of fire is not random**—it follows physical principles**.**

.

Code for Implementation of Simulation of Fire Spread in a Forest using Cellular Automata

```
clc; clear; close all;
forest_size = 50;
prob_tree = 0.7;
prob_burn = 0.5;
steps = 200;
delay_time = 1;
forest = zeros(forest_size);
for i = 1:forest_size
    for j = 1:forest_size
        if rand < prob_tree
            forest(i, j) = 1;
        end
    end
end
figure('Position', [100, 100, 700, 700]);
colormap([1 1 1; 0 0.8 0; 1 0 0; 0.2 0.2 0.2]);
imagesc(forest);
title('Forest Fire Simulation - Initial Untouched Forest', 'FontSize', 14);
axis square;
axis off;
colorbar('Ticks', [0.125, 0.375, 0.625, 0.875], 'TickLabels', {'Empty', 'Tree', 'Burning', 'Burnt'});
drawnow;
pause(1);
num_ignitions = 3;
[tree_rows, tree_cols] = find(forest == 1);
if length(tree_rows) < num_ignitions
    num_ignitions = length(tree_rows);
    fprintf('Reduced ignition points to %d due to limited trees.\n', num_ignitions);
end
if num_ignitions > 0
    rand_indices = randperm(length(tree_rows), num_ignitions);
    for i = 1:num_ignitions
        forest(tree_rows(rand_indices(i)), tree_cols(rand_indices(i))) = 2;
    end
else
    fprintf('No trees to ignite. Simulation cannot proceed.\n');
    return;
end
imagesc(forest);
title('Forest Fire Simulation - Ignition Points Added', 'FontSize', 14);
axis square;
axis off;
drawnow;
pause(delay_time);
burn_history = zeros(steps, 1);
tree_count_history = zeros(steps, 1);
burnt_count_history = zeros(steps, 1);
for step = 1:steps
    new_forest = forest;
    burn_history(step) = sum(forest(:) == 2);
    tree_count_history(step) = sum(forest(:) == 1);
    burnt_count_history(step) = sum(forest(:) == 3);
    for x = 1:forest_size
        for y = 1:forest_size
            if forest(x, y) == 2
                new_forest(x, y) = 3;
                neighbors = [
                    x-1, y;
                    x+1, y;
                    x, y-1;
                    x, y+1;
                    x-1, y-1;
                    x-1, y+1;
                    x+1, y-1;
                    x+1, y+1
                ];
                valid_neighbors = neighbors(:,1) >= 1 & neighbors(:,1) <= forest_size & ...
                                  neighbors(:,2) >= 1 & neighbors(:,2) <= forest_size;
                neighbors = neighbors(valid_neighbors, :);
                for n = 1:size(neighbors, 1)
                    nx = neighbors(n, 1);
                    ny = neighbors(n, 2);
                    if forest(nx, ny) == 1 && rand < prob_burn
                        new_forest(nx, ny) = 2;
                    end
                end
            end
        end
```

```
    end
    forest = new_forest;
    imagesc(forest);
    title(sprintf('Forest Fire Simulation - Step %d', step), 'FontSize', 14);
    axis square;
    axis off;
    drawnow;
    pause(delay_time);
    if sum(forest(:) == 2) == 0
        fprintf('Simulation ended at step %d: No more burning trees.\n', step);
        burn_history = burn_history(1:step);
        tree_count_history = tree_count_history(1:step);
        burnt_count_history = burnt_count_history(1:step);
        break;
    end
end
title(sprintf('Final State - Step %d', min(step, steps)), 'FontSize', 14);
figure('Position', [800, 100, 700, 500]);
steps_completed = length(burn_history);
plot(1:steps_completed, tree_count_history, 'g-', 'LineWidth', 2);
hold on;
plot(1:steps_completed, burn_history, 'r-', 'LineWidth', 2);
plot(1:steps_completed, burnt_count_history, 'k-', 'LineWidth', 2);
hold off;
title('Forest Fire Progression', 'FontSize', 14);
xlabel('Step', 'FontSize', 12);
ylabel('Cell Count', 'FontSize', 12);
legend('Trees', 'Burning', 'Burnt', 'Location', 'best');
grid on;
remaining_trees = sum(forest(:) == 1);
burnt_trees = sum(forest(:) == 3);
total_initial_trees = remaining_trees + burnt_trees;
fprintf('\nSimulation Summary:\n');
fprintf('Initial tree density: %.1f%%\n', prob_tree * 100);
fprintf('Fire spread probability: %.1f%%\n', prob_burn * 100);
fprintf('Trees remaining: %d (%.1f%% of initial trees)\n', remaining_trees, 100 * remaining_trees / total_initial_trees);
fprintf('Trees burnt: %d (%.1f%% of initial trees)\n', burnt_trees, 100 * burnt_trees / total_initial_trees);
```

Output

IGNITION POINTS