Project 1: Controlled Burn

Andrew Yang

Project Description

This project is a simulates system of a forest fire. In the same directory, there will have some files which provide information about the trees and weather. The locations of the trees, burning time of trees will be provided in one file. The information about wind direction and rain will be provided in another file. The user needs to input those files into the program. Therefore, the system will show how the forest fire goes; where it started and where it ended. The system will analyze and show the report of how many days needed to stop the fire.

Data Structures

Use of the const char to marks for burn status of a tree (‘T’,’F’,’B’,’ ’).

'T'= a tree that is not on fire

'F'= a tree that is on fire

'B'= a tree that has already burnt down

' '= an empty patch of land

Use of the const char to marks for forecast status of a day (‘N’,’S’,’W’,’E’,’R’).

'N'= indicates the wind is blowing to the North

'S'= blowing to the South

'W'= blowing to the West

'E'= blowing to the East

'R'= Raning, no wind blowing

I create a structure named ForestInfo to store the information of the forest.

int burnDay = How many days a tree burns for

vector<vector<char>>forestVec = The forest can be any sized rectangle

vector<vector<int>>burnOutVec = The burn out can be sized in any time(int)

I create a structure named ForecastInfo to store the information of the forecast.

int day, char flag

I create a structure named WeatherInfo to store the information of the weather.

bool lasting, char flag

I creat some function to analysis the forecast the fire goes.

// Load forest data from file

void loadForestData(ForestInfo& info, const string& filePath);

// Load forecast data from file

```
void loadForecastData(vector<ForecastInfo>& info, const string& filePath);

// Get weather data on one day

WeatherInfo getWeatherInfo(const vector<ForecastInfo>& info, int day);

// Show current status of forest and forecast

void showInfo(int day, WeatherInfo weather, const vector< vector<char> >& forestVec);

// Whether the forest is still on fire on not

bool isOnFire(const vector< vector<char> >& forestVec);

// Update the count of burn day for a tree on its position

void updateBurnStatus(const vector< vector<char> >& tmpVec, ForestInfo& info);

// Simulate on a day

void simulate(ForestInfo& info, const WeatherInfo& weather);

// Fire a tree on position(i, j) if it is a tree

void fire(ForestInfo& info, int i, int j);

// Split a string line by flag

void split(const string& s, vector<string>& sv, const char flag = ' ');
```

In main function, I write 2 functions to open the files and use those void functions to analysis the forest fire, and loop to show the diagrams. Finally, print out the result.

User needs to read the README file to know how to compile and run the program.

System Functionality

For this system, user need to read the README file to know how to compile and run the program.

First, user enter “make” in this directory in order to preprocess the program. After that user enter “./main forest1File forecastFile.txt”, for example: ./main forest1.txt forecast1.txt to run the program. If you want to redirect the result into a file: “./main forest1.txt forecast1.txt > out.txt”.

When 2 of the files input into the program, the information will be analysis into my struct function which are ForestInfo, ForecastInfo, and WeatherInfo. After that, program will put all the information into my functions to do the calculations which my functions are loadForestData, loadForecastData, WeatherInfo getWeatherInfo, showInfo, bool isOnFire, updateBurnStatus, simulate, fire, split. These function distinguish and forecast where the fire goes and when to stop, as last it will calculate how many days will the fire stop and print out the result and diagrams.