

# Scenery Team no.76

k.komali 20wh1a0558 cse

E.Sushmitha 20wh1a1237 IT

M.Sonali varma 20wh1a6656 AIML

R Greeshma 20wh1a0419 ECE

E.Himavarshini 20wh5a0407 ECE-b

27/10/2021

# Introduction

- ▶ Badlands National Park is an American National Park located in South Dakota.
- ▶ The Lakota people call this region "badlands" long ago because of its rocky terrain lack of water and extreme temperatures.
- ▶ Here, an amateur photographer capturing the photos who is very particular at lighting conditions.

# Approach

- ▶ Taking the number of pictures and time as input.
- ▶ The output tells us if we can capture the pictures in a day.
- ▶ Approaching the idea of using combinations.

# Learnings

- ▶ LaTeX
- ▶ To create Git lab repository.
- ▶ Understood how to approach the problem.
- ▶ Learn't uses of different functions.

# Challenges

- ▶ Had trouble using LaTeX - Faced difficulty in running program and how to install packages.
- ▶ Had trouble with understanding the second half of the problem statement.
- ▶ Faced difficulty in approaching the problem.
- ▶ Using functions in program.

# GIT Repo

## ► Screen Shot of Repo.

The screenshot shows a web browser window with the GitLab repository page for 'Team no.76 Scenery'. The browser's address bar shows the URL 'gitlab.com/Sonali.M/team-no.76-scenery'. The GitLab interface includes a sidebar with navigation options like 'Project information', 'Repository', 'Issues', 'Merge requests', 'CI/CD', 'Security & Compliance', 'Deployments', 'Monitor', 'Infrastructure', 'Packages & Registries', 'Analytics', 'Wiki', 'Snippets', and 'Settings'. The main content area displays a table of repository files and their last commit messages. At the top of the main area, there are buttons for 'Set up CI/CD', 'Configure Integrations', and 'Add Security Testing'. Below the table, there is a section for 'README.md'.

Name	Last commit
21st_oct.pdf	Upload New File
Python_code.py	this file contains Greeshma's code
README.md	Initial commit
Scenery	This file contains project code
himavarshini.py	Upload New File
komali_s_code.py	Upload New File
present1.pdf	Replace present1.pdf
scenery.py	This file contains project code
sushmitha_s_code.py	Upload New File
team1.pdf	Upload New File

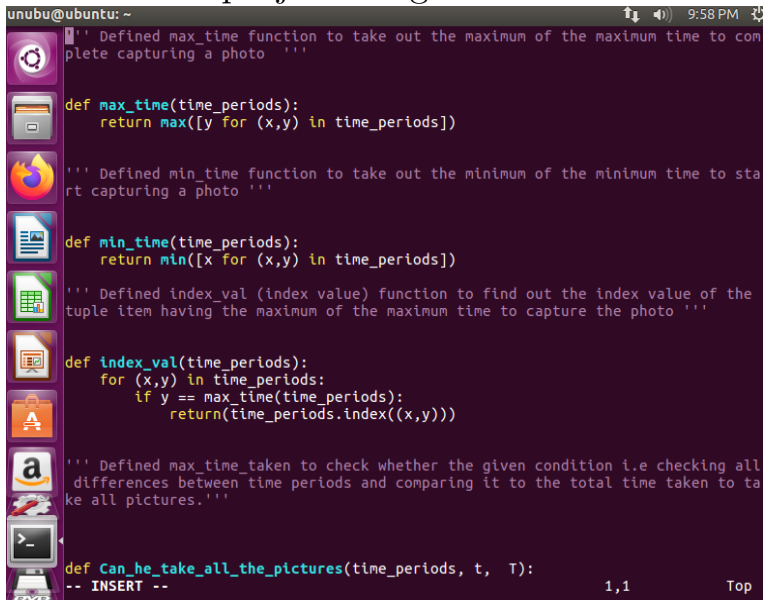
Below the table, there is a section for 'README.md'.

# Statistics

- ▶ Number of lines of code:50.
- ▶ Number of Functions:8.

# Demo picture

- ▶ Demo of the project image.



A terminal window titled 'unubu@ubuntu: ~' with a system clock showing 9:58 PM. The terminal displays Python code for a project demo. The code includes comments, function definitions for 'max\_time', 'min\_time', 'index\_val', and 'Can\_he\_take\_all\_the\_pictures', and a placeholder for an 'INSERT' statement. The terminal has a dark purple background and a sidebar on the left with various application icons.

```
unubu@ubuntu: ~
''' Defined max_time function to take out the maximum of the maximum time to complete capturing a photo '''

def max_time(time_periods):
    return max([y for (x,y) in time_periods])

''' Defined min_time function to take out the minimum of the minimum time to start capturing a photo '''

def min_time(time_periods):
    return min([x for (x,y) in time_periods])

''' Defined index_val (index value) function to find out the index value of the tuple item having the maximum of the maximum time to capture the photo '''

def index_val(time_periods):
    for (x,y) in time_periods:
        if y == max_time(time_periods):
            return(time_periods.index((x,y)))

''' Defined max_time_taken to check whether the given condition i.e checking all differences between time periods and comparing it to the total time taken to take all pictures.'''

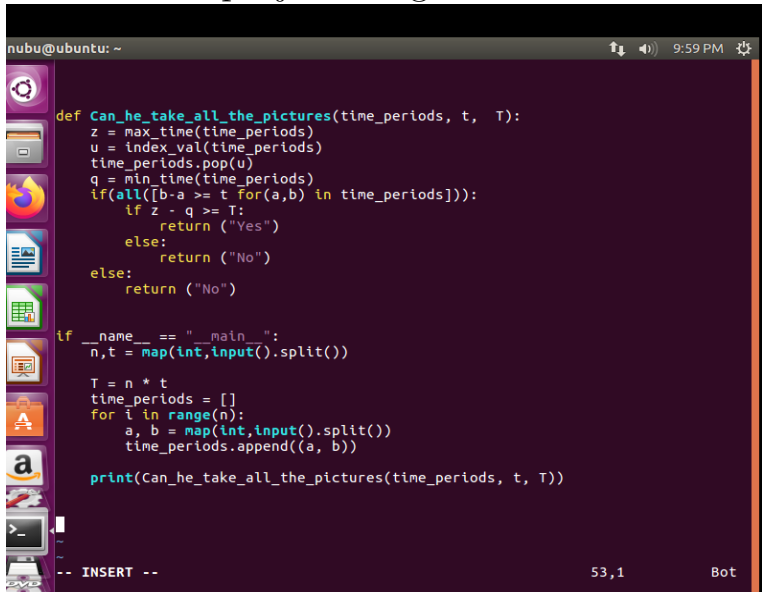
def Can_he_take_all_the_pictures(time_periods, t, T):
    -- INSERT --
```

1,1 Top



# Demo picture

- ▶ Demo of the project image.



A terminal window titled 'nubu@ubuntu: ~' with a system clock of 9:59 PM. The window displays Python code for a function 'Can\_he\_take\_all\_the\_pictures' and its main execution block. The code uses a list 'time\_periods' to store intervals and checks if a given time 't' can accommodate all intervals. The terminal has a dark purple background and a sidebar with application icons on the left.

```
nubu@ubuntu: ~
def Can_he_take_all_the_pictures(time_periods, t, T):
    z = max_time(time_periods)
    u = index_val(time_periods)
    time_periods.pop(u)
    q = min_time(time_periods)
    if(all([b-a >= t for(a,b) in time_periods])):
        if z - q >= T:
            return ("Yes")
        else:
            return ("No")
    else:
        return ("No")

if __name__ == "__main__":
    n,t = map(int,input().split())

    T = n * t
    time_periods = []
    for i in range(n):
        a, b = map(int,input().split())
        time_periods.append((a, b))

    print(Can_he_take_all_the_pictures(time_periods, t, T))

-- INSERT --
```

53,1 Bot

THANK YOU!