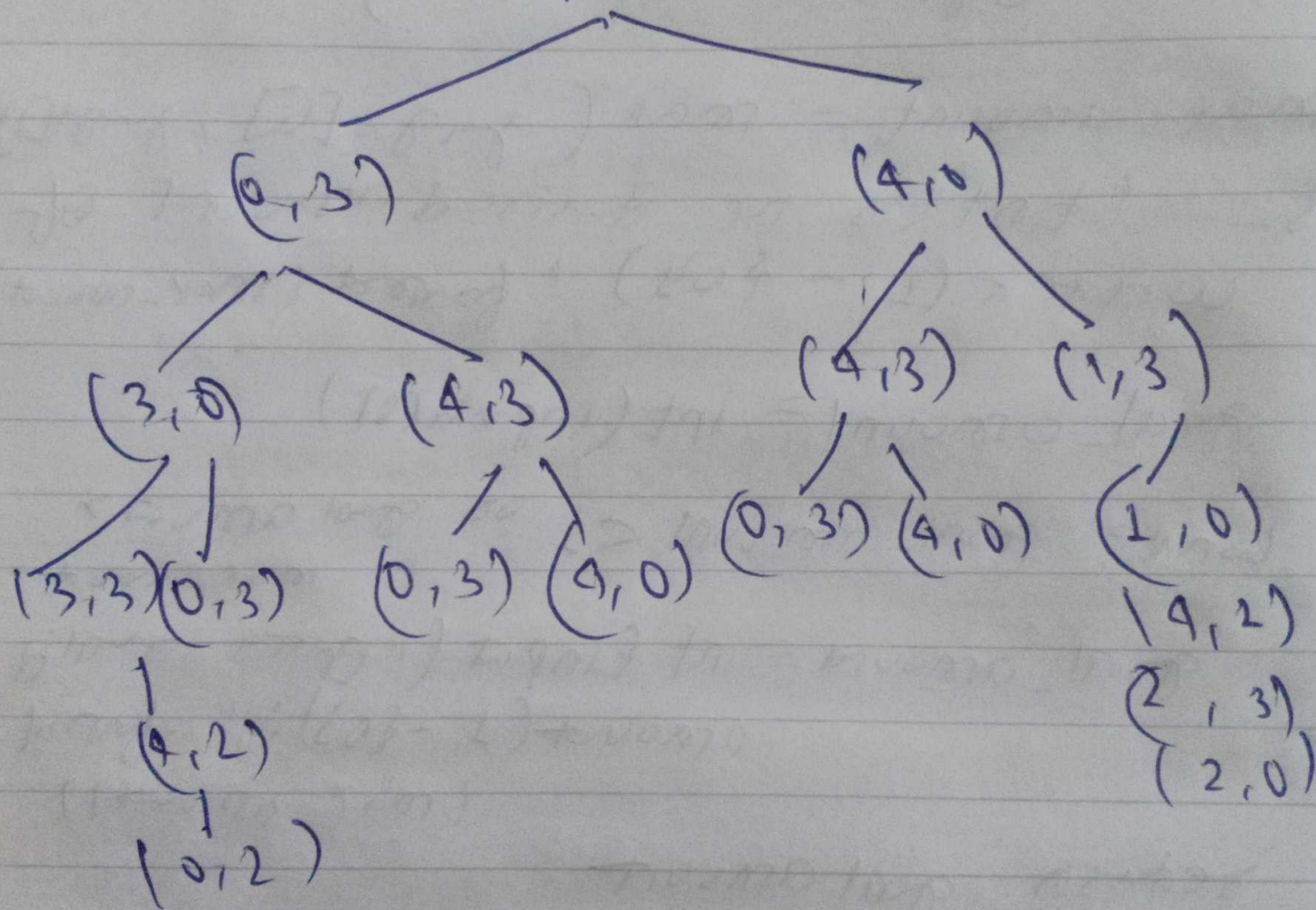


2lt of war into 4lt  
neg

death into ~~(0,0)~~ (2,0)

~~(3,3)~~ ~~(3,0)~~

(0,0)  
4lt 3lt





Samir Kumar  
LBM18CS091

13/11/20

AI lab test 1

Samir Kumar

(2) → import collections

def main():

"""

main function

"""

starting\_node = [[0, 0]]

jug1 = get\_jug1()

goal\_amount = get\_goal(jug1)

check\_dict = {}

is\_depth = {}

is\_depth = get\_search\_type()

search (starting\_node, jug1, goal\_amount,  
check\_dict, is\_depth)



Samir Kumar

LB18 CS091

13/11/20

AI lab tent 1

Samir Kumar

```
def get_index(node):
```

```
    """
```

```
    returns a key value for a given node
```

```
    node: a list of two integers representing  
    current state of the graph
```

```
    """
```

```
    return pow(7, node[0]) * pow(5, node[1])
```

```
def get_reach_type():
```

```
    """
```

```
    Returns True for DFS, False  
    otherwise
```

```
    """
```

```
    s = input("Enter 'b' for BFS, 'd' for  
              DFS: ")
```

```
    s = s[0].lower()
```



Samir Kumar

IBM18C5091

13/11/20

AI/Job text 1

Samir Kumar

while s1 != 'd' and s1 != 'b';

s = input("the input is not valid!  
enter 'b' for Bf,  
'd' for DFS;")

s = s[s].lower()

return s == 'd'

def get\_fuga():

" "

Return a list of two integers representing  
volumes of the fuga. Take volume of  
the fuga as an input from the user.

" "

temp = int(input("Enter first fuga  
volume (>1): "))

while temp < 1:



AI lab text 1

farris kumar  
LBM18(509)

farris kumar

13/11/20

```
temp = int(input("Enter a valid amount  
(> 1): "))
```

```
jug1.append(temp)
```

```
temp = int(input("Enter second  
jug volume (> 1): "))
```

```
while temp < 1:
```

```
temp = int(input("Enter a valid  
amount (> 1): "))
```

```
jug1.append(temp)
```

```
return jug1
```

```
def def get_jug1(jug1):
```

```
"""
```

Return desired amount of water  
taken on input from jug1.



def \_\_init\_\_(self, a, b):  
 self.a = a  
 self.b = b

volume = volume of the jug

print("Receiving the desired amount  
of the water...")

max\_amount = max(jug1, jug2)

s = "Enter the desired amount of  
water (1 - %d) : " % max\_amount

goal\_amount = int(input(s))

while goal\_amount < 1 or goal\_amount > max\_amount:

goal\_amount = int(input("Enter a valid  
amount (1 - %d) : " % max\_amount))

return goal\_amount