Lumberjack

Team 5

- Shriram Ghadge (180010015)
- Vibhuti Raman (180010040)
- Vipul Nikam (180010041)

Algorithm

- 1st we took total number of trees, total time available, grid size.
- We used datatype Struct(tree) to store all given properties .
- It Contains some more values like cut value of tree which is initially set to 0.

Algorithm

- We used 2D array to store x,y value to make algorithm time efficient.
- The x , y positions where tree is not there (nulltree) we gave that array point value of 0 or -1.
- And currentX, currentY are values of current position of lumberjack.

Working Process :

- Lumberjack passes its current location to function neibhour( ) which will give instruction to lumberjack in which direction he should cut the tree so that profit will bw maximum.

Working Process :

- It will calculate profit of most profitable tree (Which is stored in starProfit in our case) based on the propertie that it should be inversly proportional to time and directly proportional to profit

Working Process :

- Time includes total time to rich the tree plus time to cut it.

Profit :

CASE	PROFIT
1	49
2	18
3	82
4	249,005
5	338,160
6	283,109

CASE	PROFIT
7	259,673
8	370,720
9	179,952
10	256,160
11	235,695
12	2,573,306

Profit :

CASE	PROFIT
13	1,734,288
14	2,458,067
15	1,928,866
16	1,334,402
17	1,649,320
18	2,199,655

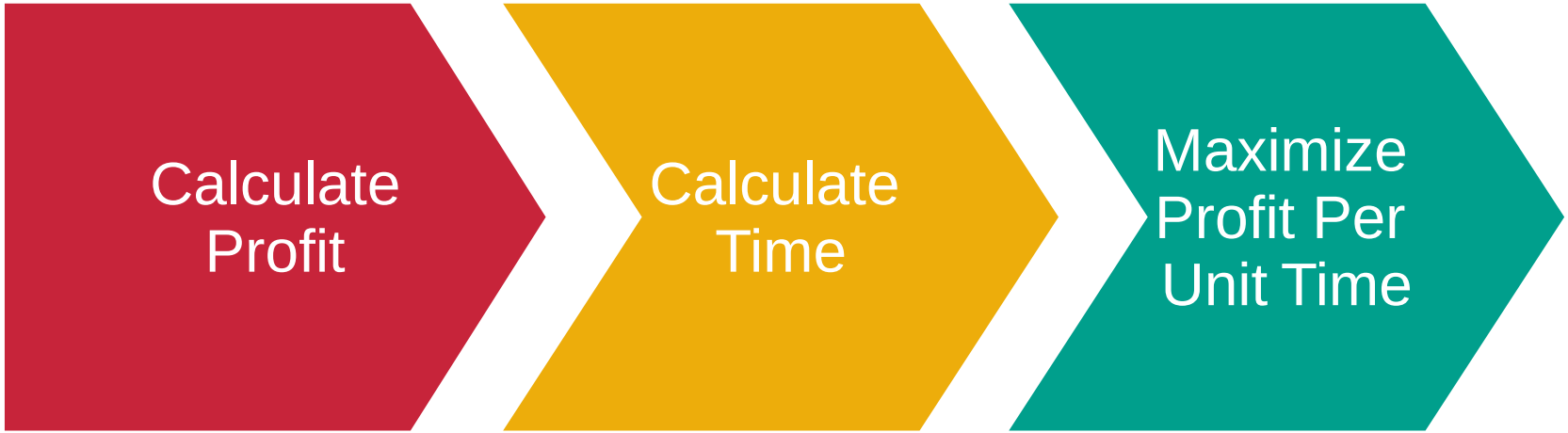
CASE	PROFIT
19	3,598,497
20	1,641,719
21	1,254,734
22	9,899,982
23	12,361,425
24	12,987,461

Profit :

CASE	PROFIT
25	9,372,870
26	14,014,040
27	10,173,672
28	9,040,850
29	24,325,416

CASE	PROFIT
30	9,283,619
31	11,571,580

Highlights :


Calculate Profit
Calculate Time
Maximize Profit Per Unit Time

Thanks