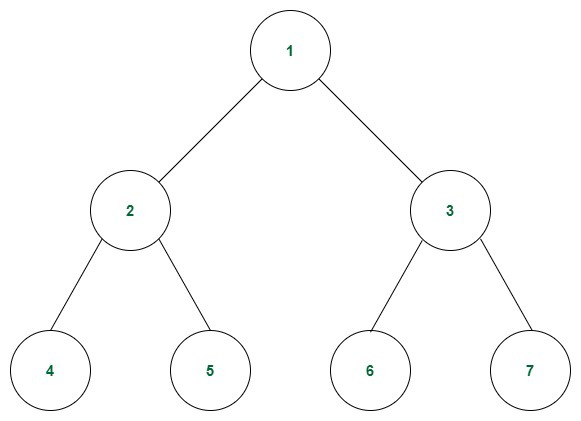
**Theoretical Assignment 1**

Question 1

Divyansh Mathur, 210357

1. **To design an algorithm in O(n3log(m)) complexity to find the final wealth of each company after m months.**

*Company hierarchy:*



Where 1 is root company and 4,5,6,7 is leaf company.

*Observation:* Consider a node which is neither root nor leaf company i.e. it has both parent and child. Let its wealth after certain years be **W** and its parent wealth be **W\_parent**. After 12 months or 1 year **W\_new** will be **211 W + 210 W\_parent** because it gives half of its wealth to its child and receives quarter of its parent wealth.

For root node it will be simply 211W.

And for leaf node it will be 210 W\_parent+ 212W

So if we are given m months it will have y=floor(m/12) years whose wealth is given by above and k=m%12 months still remaining after we have wealth after y years just multiply each node wealth by 2k to get the final wealth

*Designing Algorithm:* Since the new wealth of any node depends either only on itself or on parent and itself, we define a column matrix *Wy* of N rows such that ith row depicts wealth of **i+1** node after y years.

*#numbering is done as shown in figure starting from 1*

|  |  |  |  |
| --- | --- | --- | --- |
| ***Wy+1***  **N x 1** |  |  | ***Wy***  **N x 1** |
| Wy+1 [0] |  |  | Wy [0] |
| Wy+1 [1] |  |  | Wy [1] |
| Wy+1 [2] |  |  | Wy [2] |
| Wy+1 [3] |  |  | Wy [3] |
| . |  |  | . |
| . |  |  | . |
| Wy+1 [n-1] |  |  | Wy [n-1] |

Since we know the relationship between Wy+1 and Wy for each node we can set a relationship between these column matrices.

**NOTE:** There will be L=(N+1)/2 leaf nodes and 1 root node

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ***Wy+1*** |  |  |  | ***M*** |  |  |  |  |  |  |  | ***Wy*** |
| N x 1 |  |  |  | N x N |  |  |  |  |  |  |  | N x1 |
| Wy+1 [0] |  | 211 | 0 | 0 | 0 | . | . | 0 |  |  |  | Wy [0] |
| Wy+1 [1] |  | 210 | 211 | 0 | 0 | . | . | 0 |  |  |  | Wy [1] |
| Wy+1 [2] |  | 210 | 0 | 211 | 0 | . | . | 0 |  |  |  | Wy [2] |
| Wy+1 [3] | = | 0 | 210 | 0 | 211 | . | . | 0 |  | X |  | Wy [3] |
| . |  | . | . | . | . | . | . | 0 |  |  |  | . |
| . |  | . | . | . | . | . | . | 0 |  |  |  | . |
| Wy+1 [n-1] |  | 0 | 0 | 0 | 0 | . | . | 212 |  |  |  | Wy [n-1] |

Last L nodes will not have 211 but they’ll have 212 because they are leaf nodes and they don’t distribute their wealth.

**Wy+1 = M x Wy**

Therefore,

**Wy=My x W0**

(Where W0 denotes the initial wealth of each company)

Now, Wy can be found by Matrix exponentiation.

*Pseudocode for Matrix exponentiation:*

MULT( matrix a, matrix b){

    for i from 0 to n-1

        for j from 0 to n-1

            for k from 0 to n-1

                ANSWERMATRIX+= a[i][k] \* b[k][j]

    return ANSWERMATRIX

}

Power(M, y){

    if (y=0) return 1;

    else {

        𝑡𝑒𝑚𝑝 = Power(M, y/2);

        𝑡𝑒𝑚𝑝 = MULT(𝑡𝑒𝑚𝑝 , 𝑡𝑒𝑚𝑝);

        if (y mod 2=1 ) 𝑡𝑒𝑚𝑝 = MULT(𝑡𝑒𝑚𝑝 , y) ;

        return 𝑡𝑒𝑚𝑝;

    }

}

1. **To analyse the time complexity of your algorithm and briefly argue about the correctness of your solution.**

*Time Complexity:*

Matrix exponentiation: Power function is called almost log(y) {y is years} times and each time it is called it calls for 1 or 2 matrix multiplications, each matrix multiplication takes N3 time since the size of matrix M is N x N. This takes N3 log y time order of N3 log m, *m=12\*y*

After the above step we also multiply 2m%12 to each node, this takes O(N) time.

Therefore total time complexity is O (N3 log m + N ) **= O(N3 log m).**

*Proof of Correctness:*

**Theorem**: At the end of y years wealth of a non-leaf, non- root node is given by

**Wy=My x W0**

Where W0 is the initial wealth of each node, and M is the Matrix discussed above.

**Proof:**

W0 = M0 x W0

W0= W0

Assume Wy is correctly given as My x W0, then using the discussed Observation {

Consider a node which is neither root nor leaf company i.e. it has both parent and child. Let its wealth after certain years be **W** and its parent wealth be **W\_parent**. After 12 months or 1 year **W\_new** will be **211 W + 210 W\_parent** because it gives half of its wealth to its child and receives quarter of its parent wealth.

}

We can say ,

Wy+1 = M x Wy

Replacing Wy  with My x W0 we get

Wy+1 = My+1 x W0

Hence Proved.

1. **Consider the case of a single company (i.e. only root) in the tree. Give a constant time solution to find the final wealth after m months.**

Given single node root. It has no parent, no child. Therefore every month its wealth just doubles. Therefore after m months its wealth would be 2m times its original wealth. Assuming 2m can be calculated in constant time we can simply return 2m x W0 where W0 is the initial wealth.