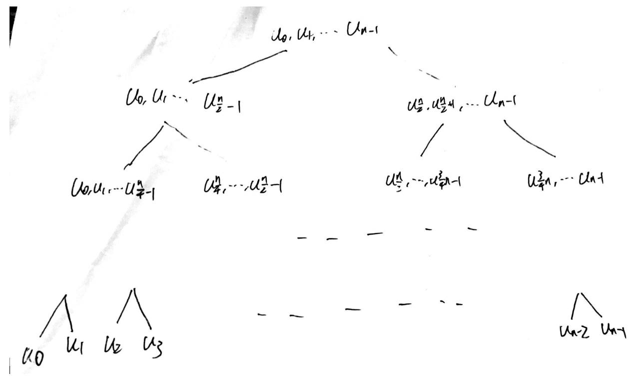
Ex 1. PART I. 1. The graph is shown below



2. Fisrst, see $M_{0,j} = \prod_{l=0}^{0} m_{j+l} = m_j$

Next, see $M_{i+1,j} = \prod_{l=0}^{2^{i+1}-1} m_{j2^{i+1}+l} = \prod_{l=0}^{2^i-1} m_{2j2^i+l} \times \prod_{l=2^i}^{2^{i+1}-1} m_{2j2^i+l} = M_{i,2j}(\prod_{l=0}^{2^i-1} m_{(2j+1)2^i+l}) = M_{i,2j}M_{i,2j+1}$

3. $M_{i,j}$ should lie on the i_{th} layer from the bottom, and j_{th} node from the left

4. (a) The algorithm is shown below.

Algorithm 1 Make Subproduct

Input: $n-2^k$,$\{u_0, u_1, ...u_{n-1}\}$
Output: Subproducts $M_{i,j}$

for $i \leftarrow 0 \rightarrow n-1$ **do**
    $M_{0,i} \leftarrow X - u_i$
end for
for $i \leftarrow 1 \rightarrow k$ **do**
    for $j \leftarrow 0 \rightarrow 2^{k-i}-1$ **do**
        $M_{i,j} \leftarrow M_{i-1,2j}M_{i-1,2j+1}$
    end for
end for

(b) The algorithm is shown below.

Algorithm 2 Fast Multipoint Evaluation

Input: P,$n-2^k$,$\{u_0, u_1, ...u_{n-1}\}$, subproducts $M_{i,j}$
Output: Evaluation of $\{P(u_0), P(u_1), ..., P(u_{n-1})\}$

function DIVIDEDOWN(f, k, i)
    if $n = 1$ **then**
        return f
    end if
    $fleft \leftarrow f \bmod M_{k-1,2i}$
    $fright \leftarrow f \bmod M_{k-1,2i+1}$
    $leftset \leftarrow DIVIDEDOWN(fleft, k-1, 2i)$
    $rightset \leftarrow DIVIDEDOWN(fleft, k-1, 2i+1)$
    return $\{leftset, rightset\}$
end function
return $DIVIDEDOWN(P, k, 0)$

5. (a) If the tree has only one layer, which means that f will just be a constant, and returning f directly is surely true.

Suppose that the values is correct for $k-1$ layers, consider the situation on k_{th} layer.

Because we are doing modulus operation, so the true function P can be written as

$$P = qleft(u_i)M_{k-1,0} + rleft(u_i)$$

or

$$P = qright(u_i)M_{k-1,1} + rright(u_i)$$

It can be seen that the qM term will be 0 for the value we want to use and so the true value will be in the r term, which we evaluate using a $k-1$ layer tree, thus the correctness is proven.

(b). The recurrance relation can be written as $T(n) = 2T(n/2) + M(n)$, thus the complexity is $\mathcal{O}(M(n)\log n)$

PART II 2. $m' = \sum_{i=0}^{n-1}(X-u_i)'\frac{m}{X-u_i} = \sum_{i=0}^{n-1}\frac{m}{X-u_i}$, and because all terms with $(X-u_i)$ will be zero in m', the only term remains is $\frac{m}{X-u_i} = 1/s_i$
3. The algorithm is given below

Algorithm 3 Fast Interpolation

Input: $n-2^k$,$\{u_0, u_1, ...u_{n-1}\}$, $y = \{P(u_0), P(u_1), ..., P(u_{n-1})\}$, subproducts $M_{i,j}$
Output: P

```
function multiUp(f, k, i)
    if n = 1 then
        return y
    end if
    fleft ← MULTIUP(fleft, k − 1, 2i)
    fright ← MULTIUP(fleft, k − 1, 2i + 1)
    return fleft × M_{k,1} + fright × M_{k,0}
end function
return MULTIUP(P, k, 0)
```

---

5. The recurrance relation can be written as $T(n) = 2T(n/2) + M(n)$, thus the complexity is $\mathcal{O}(M(n)\log n)$

6. The calculation is correct, but there might need much space to do this job.

Ex 2. 2. Let us check this dialogue sentence by sentence, an we pay more attention on the logic of S_1

- S1 : "Im not surprised, I knew you couldnt know!"

  It means that in S_1's mind, however you decompose his sum, you can never have a product that can be uniquely factored in this scenario.

  This sentence can exclude most of the possibilities of the sum. First, every number larger than 54 can be excluded, because those sums can all be possibly decomposed as $53 + x$, and if S_2 has $53x$, he can guess it because you can only have 53 as an individual factor.

  Next, all sums that can be decomposed to two primes can be excluded. There are mainly two cases: $2 + prime$ or $oddprime + oddprime$. For $2 + prime$, we can exclude $4, 5, 7, 9, 13, 15, 19, 21, 25, 31, 33, 39, 43, 45, 49$. And for $oddprime + oddprime$, actually we can exclude all even numbers, based on the famous Goldbach's conjecture (we can assume that it is correct in such a small range). In that case, what we have now are $11, 17, 23, 27, 29, 35, 37, 41, 47, 51, 53$

  Consider the case of 51, it is possible that the two numbers are 17 and 34. In that case the product will be 2×17^2, and the only possibility is 17 and 34, so it can also be excluded.

- S2 : "Uhm ... so now I know ..."

  When he heard that what S_1 said, he will compress the range of sums into what we concluded. And as he knew it, it means that although his product has multiple kinds of decompositions, only one of the possible sum falls in this set.

- S1 : "So do I!"

  It means that the answer of S_2 will not confuse S_1. In this way we can exclude those whoe can be decomposed to $2^n + p$(p is prime) in more than one ways. Because when S_2 has $2^n p$ and he gets what S_1 said, he know that the only possibility is $2^n + p$, but S_1 still is not able to decide from $2^{n_1} + p_1$ and $2^{n_2} + p_2$. Then the remaining are $17, 29, 41, 53$, and now we see it one by one.

Consider the case $29 = 2 + 27 = 4 + 25$, and in both cases S_2 can then know while S_1 still cannot judge.

$41 = 4 + 37 = 10 + 31$, same

$53 = 6 + 47 = 16 + 37$, same.

So the only case is $17 = 13 + 4$, the two numbers are 13 and 4

3. In fact the word "if they collide they all go reversely" is of no use.Because all ants are the same, the situation should be the same as the situation when the cable is wide enough that both ants can pass each other when they meet. So the maximum total time should still be 1s