* 1. Which does this approach work

* 1. Change number n to base 2, it will be shown as which is 0 or 1.

=

So we only need to calculate matrices, each one is the square of the previous one.

The total steps are, time complexity is



Using 3 real numbers multiplications and 5 real numbers additions.

Using 2 real numbers multiplications and 3 real numbers additions

Using 3 real numbers multiplications we could calculate

Using 2 real numbers multiplications we could calculate

So using 5 multiplications only.

2. 000 001 010 011 100 101 110 111

000 100 010 110 001 101 011 111

So change the index to binary, reverse the digits.

The new index is the permutation of the leaves.

1. Let the edge length as the node value

int transform(Node<Integer> root) {

if (root == null) {

return 0;

}

int leftValue = transform(root.getLeft());

int rightValue = transform(root.getRight());

int maxValue = Math.max(leftValue, rightValue);

if (leftValue != 0) {

root.getLeft().setValue((int) root.getLeft().getValue() + maxValue - leftValue);

}

if (rightValue != 0) {

root.getRight().setValue((int) root.getRight().getValue() + maxValue - rightValue);

}

return maxValue + root.getValue();

}

1. Sort items by polishing time by descend. Then process them one by one.

Assume the previous solution is not the optimal one. In order to achieve the optimal one, we should swap at least one time.

, would have no effect on the completion time. But may cause completion time delay or have no change.

So swap will let the solution same or worse.

Thus is the optimal solution.

1. Start from any vertex and set it as selected area. Choose the shortest edge to the selected area. Add the edge and vertex into the selected area. Repeat until all vertices are added into selected area.
   1. Set Loololong as START
   2. Set first house start from START as FIRST;
   3. Go straight until reaching the position which is 5 km far away from FIRST;
   4. Set a base station on this position;
   5. Move forward 5km and set current position as START
   6. Repeat step 2 to 5 until reaching Goolagong;
2. 1. Find the people in the people list whose known number or unknown number is less than 5.
   2. Remove this one from people list and Pairs list.
   3. Repeat finding the people and remove him/her until everyone’s known number and unknown number is greater than or equal to 5.
3. 1. If start by connecting the closest pair of black and a white dot, Connection A would be one of the possibility. But the total length is 4 which is larger than the optimal length 2 as connection B.

B

A

* 1. The optimal solution:

Select the left most white node and left most black node.

Connect and ignore them.

Repeat

Create an empty **List** which length is n.

For task i = 1 to n:

If from **List[0]** to **List[i-1]** , one or more are empty:

Put into the right most empty one.

Else:

If is less than all values:

Add to **totalPenalty**

Else:

Put into the one has minimum value.

Add that minimum value to **totalPenalty**

1. BookList = [10]

While bookId in sequence:

If bookId in BookList:

Continue

Else:

Add bookId into BookList

If BookList is full:

Go to library and the books in Booklist

Clear BookList

1. Select 10 longest areas which do not need to be covered and not include the first and last stall.

Then cover the other 11 parts.



len1 = len(string1)  
len2 = len(string2)  
while len1 > 0 and len2 > 0:  
 if string1[len1 - 1] == string2[len2 - 1]:  
 romve the letter on the position (len2-1) from the string2  
 len1 -= 1  
 len2 -= 1  
 else:  
 len2 -= 1  
if len1 == 0:  
 return string2  
if len2 == 0:  
 return None