COP 3503 3/28/24

(1) Grade Calc Webcourses (E1, E2, P1, P2, max(RP1, PR2, RP3))

48%

A = 75% B = 60% C = 45%

exam .65 progs .21 rps .09

WC $w_1 p_1 + w_2 p_2 + w_3 p_3$

Cur $w_1' p_1 + w_2' p_2 + w_3' p_3$

currents weights

$\frac{40}{48}$ $\frac{5}{48}$ $\frac{3}{48}$

$\frac{5}{6}$ $\frac{1}{16}$

$< 30\% \rightsquigarrow$ withdrawing

Divide + Conquer

(1) Closest Pair of Points - Typed

(2) Repeated Shuffling/Permuting - UCF HS Prog Contest 2017

(3) # of paths in a graph of a fixed length - Type eventually

Closest Pair of Points

Input: n pts Cartecian Plane

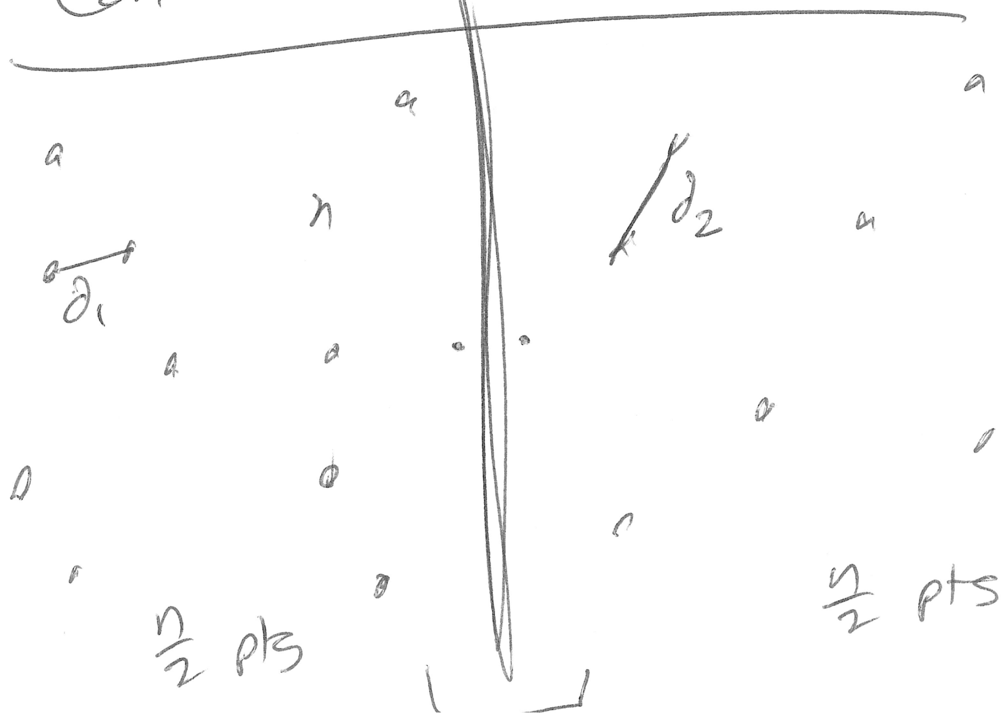
Output: Distance btw the 2 closest pair of pts.

Alg #1

$\boxed{O(n^2)}$

```
res = ∞
for (i=0; i<n; i++)
    for (j=i+1; j<n; j++)
        res = min(res, dist(pt[i], pt[j]))
```

Can Div+Conq Help?

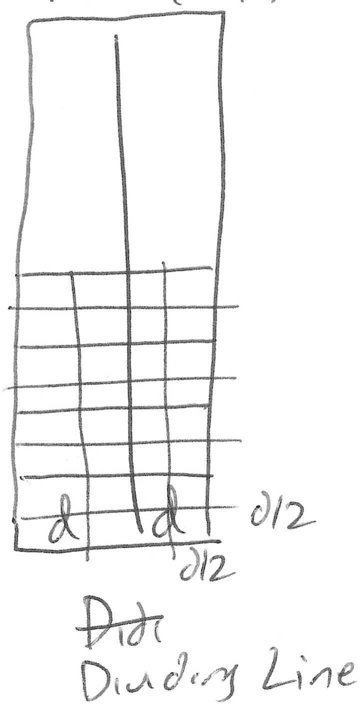


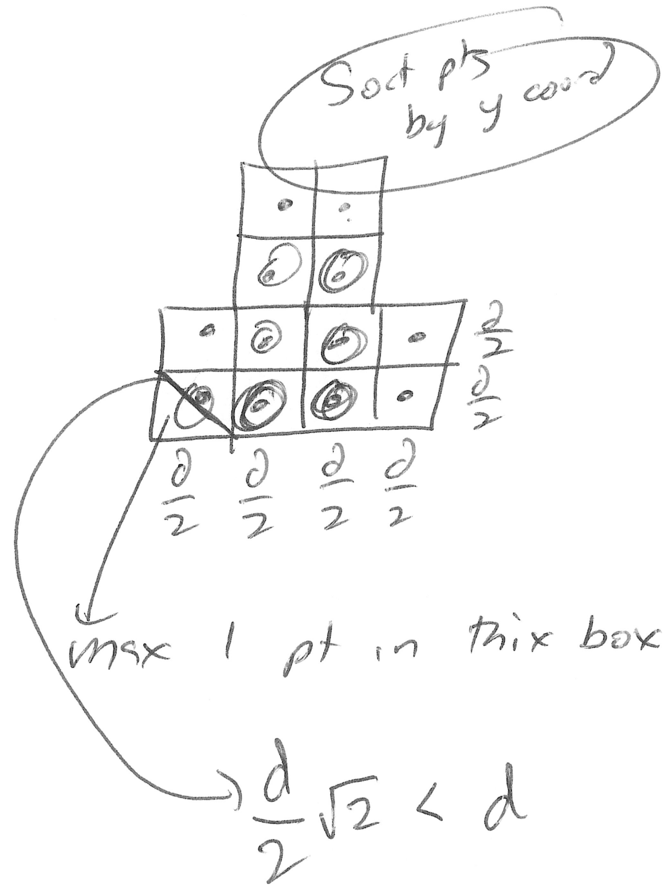
if we do $\min(\partial_1, \partial_2)$ error is maybe 2 pts on diff sides are closer.

IDEA

Still do recursion but can we only check "a few points" near the dividing line?

Let $d = \min(d_1, d_2)$





① Sort strip by y coordinate

② for each pt just check the next 7 pts after it in array

extra time $O(n)$

Alg

① Sort by x
② Recursively solve left half
③ Recursively solve right half
④ Get min d, get all pts with d split

(5) Sort These by y

(6) Double for loop (n, 7) + set closest pair

(7) Ans is best of 3 options.

2→1→3→2
4→4→4→4
3→2→1→3
1→3→2→1

Every Day I'm Shuffling

Filename: `shuffling`

There is a party rock in the house tonight, so naturally everybody is having a good time. But sitting in the corner, Redfoo and Sky Blu are shuffling their deck of cards and simply proclaiming, "Every day I'm shuffling."

Redfoo and Sky Blu have a new deck of n cards which originally are in order where the i^{th} card from the top has the number i written on it (1, 2, …, n-1, n). Just as their song is repetitive, so are Redfoo and Sky Blu's shuffling efforts. They have been meticulously shuffling their deck once a day for many days now. The process they use is simple. They deal out the n cards from left to right from the top, maintaining the order of the cards. The leftmost card (originally the top card) has position 1, while the card on the right has position n. They then refer to their special permutation of the numbers 1 through n (p_1, p_2, …, p_{n-1}, p_n), which tells them in what order they pick up each card. The i^{th} card they pick up should be the one in position p_i. Each card they pick up is put on the bottom of the growing pile. Picking up a card does not affect the position numbers of all the remaining cards still laid out. For example, a deck of [2, 1, 3, 4] (from top to bottom) with a special permutation of [2, 3, 1, 4] will result in a new deck in the order [1, 3, 2, 4] (from top to bottom). They then repeat this process with the new order of cards the next day, still using the same special permutation from day 1.

2→1
3→2
1→3
4→4

→ OLD DECK

→ PERM

Unfortunately, after many, many days of shuffling their cards, Redfoo and Sky Blu have noticed that the numbers on each card have faded away. They wish to return their cards to their original brilliance but would not like to renumber any cards. Thus, they need your help to determine what number should be on each card given the special permutation they use and the number of days they have shuffled their deck.

The Problem:

Given the special permutation that Redfoo and Sky Blu use to shuffle their special deck of cards every day and the number of days, print the deck from top to bottom after they have shuffled it for that many days.

10^5 $k \le 10^9$

how many times to shuffle

The Input:

The input consists of multiple shuffling sprees. The first line of the input will contain a single, positive integer, t, the number of sprees that follow. Each spree will consist of two lines. The first will contain two integers, n_i and k_i ($1 \le n_i \le 100{,}000$; $0 \le k_i \le 10^9$), representing the number of cards in Redfoo and Sky Blu's special deck of cards, and the number of days they have been shuffling their deck, respectively. The second line will contain n_i integers, the special permutation of the numbers 1 to n_i which represents the pattern used to shuffle the deck on that particular spree.

→ Perm

1→3
2→1
3→2
4→4

The Output:

For each shuffling spree, output the n_i integers from 1 to n_i in the order of the deck after it has been shuffled for the k_i days in that spree using the appropriate special permutation. The j^{th} number on each line should represent the number written on the j^{th} card from the top.

Sample Input:

```
2
4 2
2 3 1 4
5 10
4 3 5 1 2
```

Sample Output:

```
3 1 2 4
1 3 5 4 2
```

If I do 1 shuffle it takes 10^5 step

If I do 10^9 shuffles == $\underline{\underline{10^{14}}}$ steps

BAD

10^8 shuffles

What if I knew the perm func for 5,000,0000 $= 5 \times 10^7$ $\left(\frac{10^8}{2}\right)$?

$1 \to 4 \to 2$

$2 \to 3 \to 5$

$3 \to 5 \to 1$

$4 \to 2 \to 3$

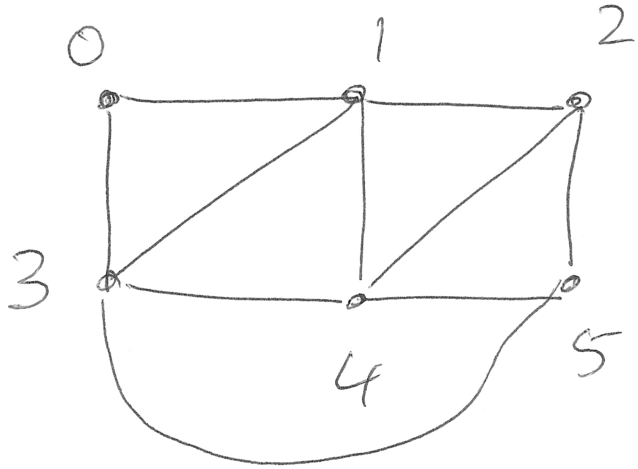
$5 \to 1 \to 4$

50mil 50mil

* perm f(perm, k)

like fast modular exponentiation

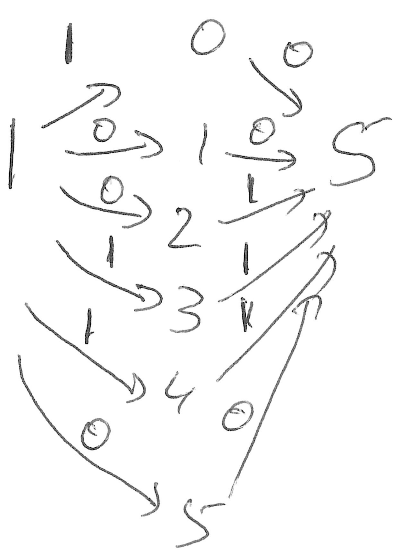
Paths in a graph of a fixed length



how many paths are there from vertex 1^{a} to vertex 5^{b} of length 2_{k}.

m[i][j] stores ans for paths of length 1.

$$
M = \begin{array}{c|cccccc}
 & 0 & 1 & 2 & 3 & 4 & 5 \\
\hline
0 & 0 & 1 & 0 & 1 & 0 & 0 \\
1 & 1 & 0 & 1 & 1 & 1 & 0 \\
2 & 0 & 1 & 0 & 0 & 1 & 1 \\
3 & 1 & 1 & 0 & 0 & 1 & 1 \\
4 & 0 & 1 & 1 & 1 & 0 & 1 \\
5 & 0 & 0 & 1 & 1 & 1 & 0
\end{array}
$$

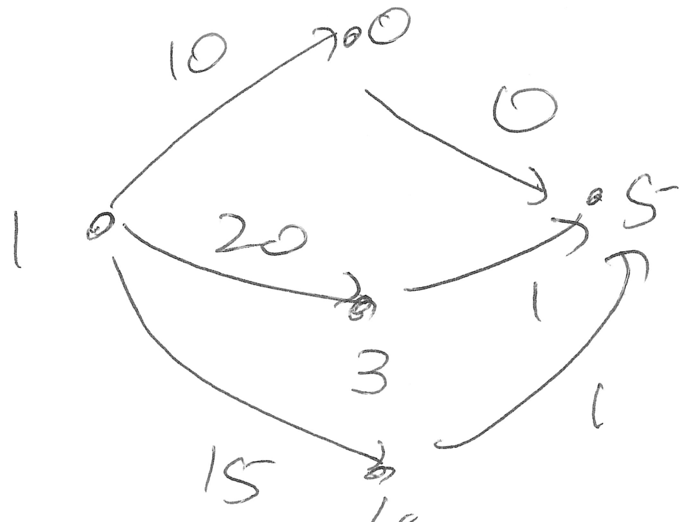


$1 \times 0 +$
$0 \times 0 +$
$0 \times 1 +$
$1 \times 1 +$
$1 \times 1 +$
0×0

$$\sum_{k=0}^{5} m[i][k] * m[k][j]$$

$$M^2 = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 \end{bmatrix}$$

$$M^{k-1} \times M$$



\# path length $k-1$

$10 * 0 +$

$\vdots$

$20 \times 1 +$

$15 \times 1 +$

$\vdots$

$= 35$

$M^k \rightarrow$ this entry $M^k[i][j]$ equals # paths of length k from vertex i to vertex j.

Fast Mat Expo (exp $\frac{1}{2}$ way) to speed this up!!!)