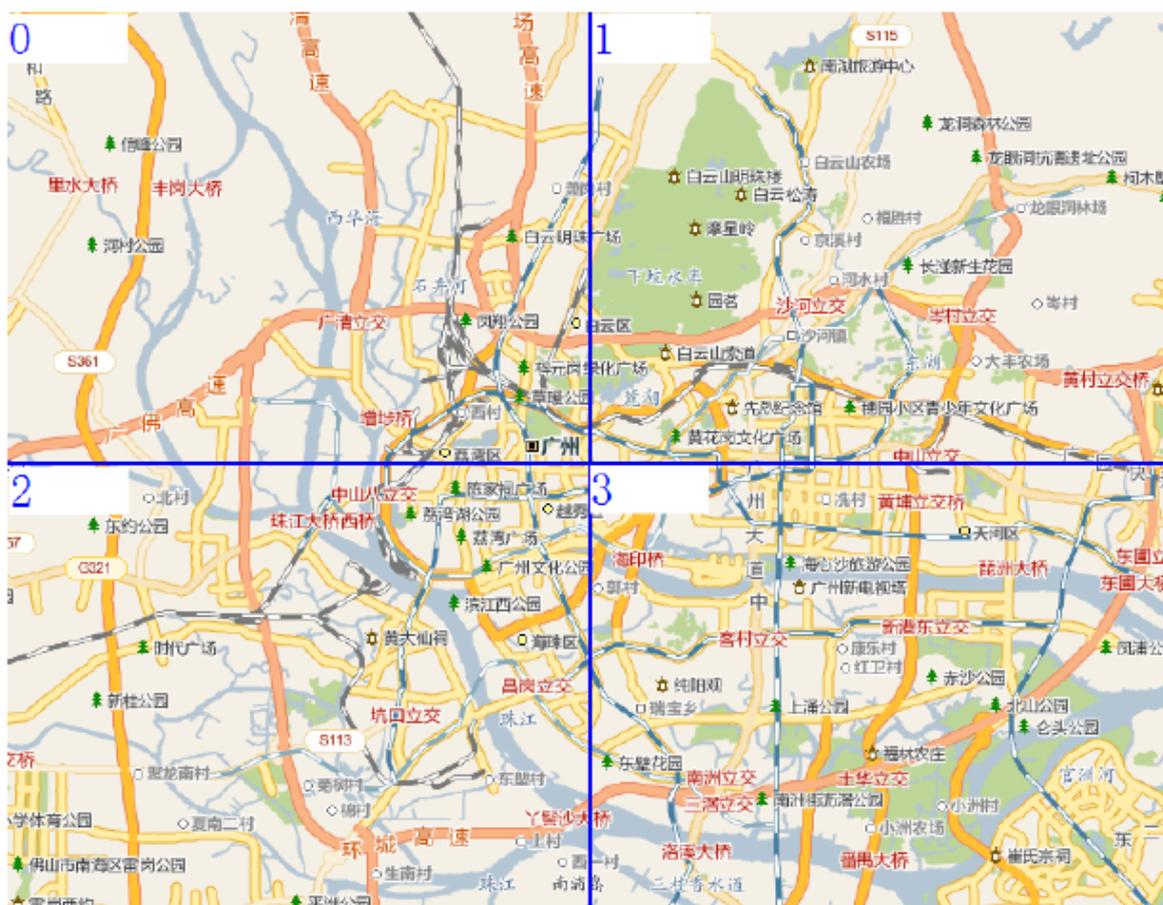


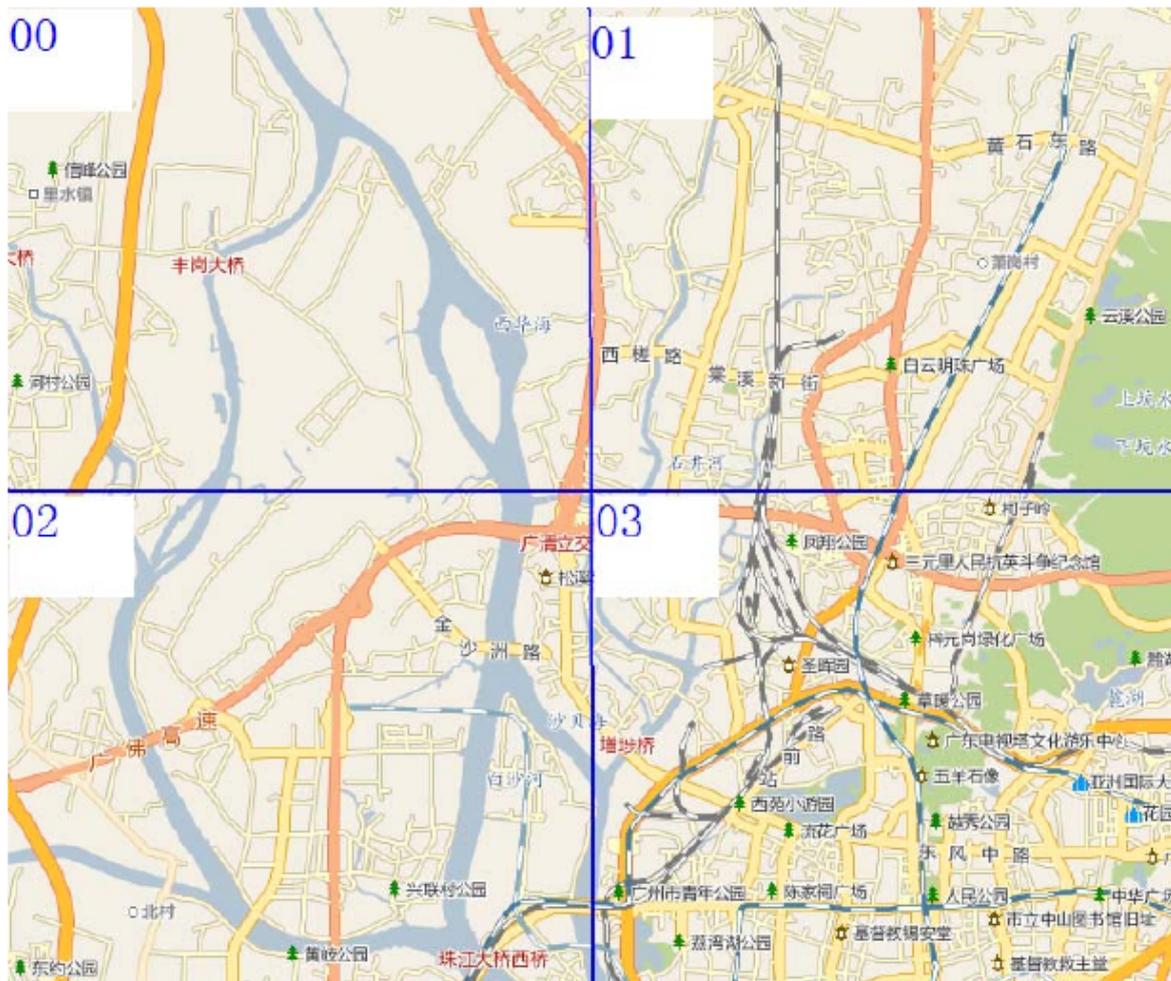
4400 Transit search

Henry decides to develop a web site, which will provide the service of transit search. But he can only get the transit data of Guangzhou, so his web site can only support the transit search of Guangzhou. We suppose Guangzhou is 10240 meters by 10240 meters. The coordinate of the top-left corner is (0,0). The coordinate of the bottom-right corner is (10240,10240). The X-axis is from top to bottom and the Y-axis is from left to right. At the beginning, four pictures of the size 10cm by 10 cm make up of the whole map of Guangzhou. They are numbered from 0 to 3. It is to say at the beginning the scale of the map is 1cm:512 meters. We call the four pictures are at level 1.



When you double-click on the map using the mouse, the map will zoom in. The pictures at next level will be shown on the screen. For example, when you double-click on the above map, picture 0 will be replaced by four pictures 00, 01, 02, 03, all of whose sizes are 10 cm by 10 cm. and the scale of the map change to 1cm:256 meters. (notice that, pictures 00,01,02,03 together describe the same area as picture 0). When you continue double-click, picture 01 will be replaced by pictures 010,011,012,013, and so on.

Now, a position's coordinate can be given by (str, x, y) . str consists of 8 characters each from 0 to 3. It describes the id of the picture which the position is located at. x and y ($0cm \leq x, y \leq 10cm$) describe the position's offset relative to the top-left corner on picture str . Notice that the X-axis is from top to bottom and the Y-axis is from left to right.



Now, the start position and end position are given as $(start, s_x, s_y)$, (end, e_x, e_y) . And some information about the bus line will be also given. First, each bus stop will be described by $(name, x, y)$, its name and its coordinate. Second, each bus line will be described by $(name_1, name_2, name_3 \dots name_k)$ which are the bus stops the bus line travels through. If the distance between the start position and end position is no more than 2000 meters, the web site will suggest walking there. Otherwise, the web site will find a bus stop whose distance is no more than 1000 meters from the start position. You can take buses to a bus stop whose distance is no more than 1000 meters from the end position. Along the way, you can change buses at any bus stop. If you can take buses according the above rules, the web site will find a route with fewest number of changing buses. If you can't take buses according the above rules, the web site will suggest taking a taxi.

Input

The input begins with a line containing an integer T , the number of test cases.

For each case, the first two lines describe the start position and the end position as followed.

Start $s_x s_y$

End $e_x e_y$

Start and *End* both contain 8 characters each from 0 to 3. $0cm \leq s_x, s_y, e_x, e_y \leq 10cm$. Notice that all the numbers in the input are integers.

The next line contains an integer n ($0 < n < 5001$), indicating the total number of bus stops in Guangzhou. The following n lines each describe a bus stop in the format:

Name x y

Name contains no more than 20 characters. $0 \leq x, y \leq 10240$.

Next comes an integer m ($0 < m \leq 100$), indicating the number of bus lines in Guangzhou.

Then following is the description of the m bus lines.

Each bus line is described as followed:

K

$Name_1 Name_2 Name_3 \dots Name_k$

K ($0 < K \leq 30$) is the number of bus stops along the bus line.

$Name_i$ is the i -th bus stop along the bus line. Notice that the bus line is bidirectional.

Output

- (1) If the distance between the start position and end position is no more than 2000 meters, print 'walk there' in a single line.
- (2) If you can take buses according to the above rule, print the fewest number of buses you have to take. For example, if you can take a bus directly to end position without changing bus line, print '1'.
- (3) Otherwise, print 'take a taxi' in a single line.

Sample Input

```
3
00000000 1 1
00001000 3 3
4
a 1 1
b 20 30
c 40 50
d 100 100
2
3
a b c
3
b c d
00000000 1 1
03231130 5 5
5
a 1 1
b 1000 1000
c 3000 3000
d 3000 4000
e 4500 4000
2
3
a b c
3
c d e
00000000 1 1
```

03231130 5 5

4

a 1 1

b 1000 1000

c 3000 3000

d 3000 4000

2

3

a b c

3

b c d

Sample Output

walk there

2

take a taxi