需要有userid cuid \_nm\_key之类的用户id随便给一个就行，反正不用。

na 端，需要xy坐标，记为search xy，和请求poi，可以是多个，逻辑是这样的：

如果seach\_xy与query\_poi\_list里面的有一个query\_poi的距离小于5公里，就是附近搜索。

例如，query\_poi\_list里面有6个query\_poi，如果这六个poi中只需要满足其中一个与seach\_xy的距离小于5km，就认为是附近搜索。

附近搜索下，把query\_poi\_list里面的query\_poi与seach\_xy的距离，重排query\_poi位置，由近及远。见case“附近搜索 两个poi附近”和“附近搜索 多个poi附近”。例如 query\_poi\_1与seach\_xy的距离为2.1km，query\_poi\_2为1.2km，query\_poi\_3为0.8km，query\_poi\_4为10.6km，那么query\_poi\_list重排为query\_poi\_3，query\_poi\_2，query\_poi\_1，query\_poi\_4。

重排query\_poi\_list后，会取query\_poi对应的推荐列表，并对推荐poi与searchxy的距离进行过滤，当距离大于8km，这个recom\_poi就会被去掉。例如有两个query\_poi: query\_poi\_2, query\_poi\_1，其对应的推荐列表如下：

query\_poi\_2：recom\_poi\_5（6.5km）, recom\_poi\_6（1.2km）, recom\_poi\_7（2.6km）, recom\_poi\_8（15km）（离线保证这个列表不会重复!）

那么过滤后，recom\_poi\_8就会被干掉。

query\_poi\_1: recom\_poi\_1（20km）, recom\_poi\_2（90km）, recom\_poi\_5（30km）, recom\_poi\_4（80km）

那么过滤后，query\_poi\_1的列表就空了。

如果经过距离过滤后，剩下的poi小于1个，也就是假设那么会找回与search xy最近的一个推荐poi，例如，两个query\_poi: query\_poi\_2, query\_poi\_1：

query\_poi\_2：recom\_poi\_5（16.5km）, recom\_poi\_6（11.2km）, recom\_poi\_7（12.6km）, recom\_poi\_8（115km）（离线保证这个列表不会重复!）

query\_poi\_1: recom\_poi\_1（20km）, recom\_poi\_2（90km）, recom\_poi\_5（30km）, recom\_poi\_4（80km）

距离过滤后两个列表都空了，那么会找回recom\_poi\_6（11.2km），因为他是最近的。

距离过滤后，然后按位置序融合这些推荐列表，重复的poi会按先出现的位置排序。例如两个query\_poi: query\_poi\_2, query\_poi\_1：

query\_poi\_2：recom\_poi\_5, recom\_poi\_6, recom\_poi\_7, recom\_poi\_8（离线保证这个列表不会重复!）

那么过滤后，recom\_poi\_8就会被干掉。

query\_poi\_1: recom\_poi\_1, recom\_poi\_2, recom\_poi\_5, recom\_poi\_4

位置序融合后：

recom\_poi\_5，recom\_poi\_1，recom\_poi\_6，recom\_poi\_2，recom\_poi\_7, recom\_poi\_8，recom\_poi\_4。

位置序融合的列表，会改写分值，用公式(list\_len - pos)/ list\_len；如：

recom\_poi\_5分值为(7 – 0)/ 7

……

recom\_poi\_4分值为(7 – 6)/ 7

改写分值后的推荐列表，就会返回给调用方。

如果seach\_xy与query\_poi\_list里面的全部query\_poi的距离大于5公里，就是非附近搜索。那么直接取query\_poi\_list的推荐列表，并直接走位置序融合，分值改写，吐出推荐列表。

PC端，需要有cityid（也就是areaid），和请求poi，可以是多个，逻辑是这样的：

query\_poi\_list的中poi的cityid与search\_city\_id做match过滤，两个id不匹配的query\_poi会被干掉。如果最后把所有query\_poi都干掉了，那么会回滚，吧所有被干掉的poi都捞回来。

然后直接取query\_poi\_list的推荐列表，并直接走位置序融合，分值改写，吐出推荐列表。