

雖然放在 1001 的位置但其實是最坑的一道題。解法應該是不唯一的。有人做出來嗎？

首先題意就非常坑：查詢分雙關鍵字和單關鍵字兩種，但是很顯然它們兩個做法並不一樣。單關鍵字的查詢任何人都會寫，放在這裏只是增加無謂的細節判斷而已。另一方面，既然所有雙關鍵字查詢的係數 P 和 Q 都是一樣的，那麼題目核心其實就是 $P=Q=1.0$ ，這裏額外讀入的 P 和 Q 同樣祇是增加無謂的細節處理而已。另外，項目和關鍵字的關聯一共 $3N$ 個（這裏+5 是因為生成數據的程序寫殘了多生成了幾個）其實和每個項目各自關聯 3 個關鍵字也沒區別。輸入裏數量極少的修改操作祇是爲了強制在綫而已，也無傷大雅。（最終版的題目裏刪去了修改操作）

於是真實的題目其實是這樣的：給 N 個項目和 N 個關鍵字，每個項目和其中 3 個關鍵字以一定權值關聯，每次查詢給出 2 個關鍵字，讓所有項目對這兩個關鍵字的權值加和從大到小排序，並輸出第 K 個。

考慮那些暴力的做法。每次都 `nth_element` 顯然不行。這題是數據結構題目，也許可以用 `BST`。首先每個關鍵字建立一個 `BST`，然後查詢的時候可以把兩個關鍵字的 `BST` 合併在一起然後查詢，這是 $O(N)$ 的。或者也可以預先把兩兩關鍵字合併構成 `BST`，這樣在一開始需要 $O(N^2)$ 的 `BST`，但是查詢很快。這兩個做法在預處理和查詢時各自有着不同的複雜度，因此考慮使用根號方法合併兩個思路。

每個關鍵字，如果其包含的內容超過 $G=\sqrt{3*N}$ 個，稱爲大組；反之稱爲小組。大組的個數不超過 G 個，因此可以在預處理的時候暴力在大組之間兩兩建立組合 `BST`，這樣兩關鍵字都是大組的查詢可以 $O(\log N)$ 解決，修改則需要 $O(G \log N)$ 。對於有小組的查詢，每次查詢的時候可以把小組的元素先插入到另一組的 `BST` 中，查詢得到結果後再刪掉剛插入的元素，複雜度是 $O(G \log N)$ 。於是總複雜度應該是預處理 $O(N \log N + NG \log N)$ ，一次查詢或修改是 $O(G \log N)$ ，加在一起是 $O((N+Q)\sqrt{N} \log N)$ 。考慮到 N 祇有 10^4 ，這是可以接受的。

最後，如果沒有修改、允許離綫的時候，我們有更好的做法：不再區分大小組，每個關鍵字建立一個 `BST`，不做其他的預處理；然後直接對所有查詢排序，把 A 、 B 相同而 K 不同的查詢放在一起處理。每次遇到新的 (A,B) 時，就把關鍵字 A 和關鍵字 B 的 `BST` 合併，把相對較小的 `BST` 暴力插入到較大的那個裏面去，全部查詢完再復原。這樣總複雜度和上面的做法是完全一樣的。爲甚麼是這個複雜度請各位自行思考。

因爲 `STL` 的 `set` 不支持查詢 K 大，所以還得自己寫一個 `SBT`。輸入比較多所以最好寫個輸入外掛。小數因爲祇有 3 位，所以可以當成 `int` 處理。所有的細節都考慮好的話，程序應該可以在 1s 內跑完所有用例（大用例還是挺水的）。之所以開到 10s 是爲了防止卡常數以及支持一下其他的做法。公開用例以及涉及到了所有的細節，所以非暴力通過明例的話肯定是可以 AC 的。