

## پروژه درس ساختمان داده

### مهلت تحویل : 2 بهمن

پروژه به صورت تک نفره تعریف شده است

صورت سوال:

امیر که اهل شیراز نیست به تازگی به شیراز آمده است.

از آنجایی که او به شهر آشنایی ندارد برای تهیه غذا از گوگل مپ استفاده می کند.

حال با توجه به اینکه سفارش دادن از رستوران دور تر زمان بیشتری را برای آماده شدن سفارش می طلبد و او گرسنه تر از این حرفاست که بتواند زمان زیادی را منتظر غذا بماند او میخواهد از بین  $k$  رستوران نزدیک به خودش دنبال رستورانی با پیشوند  $S$  بگردد (زیرا او تنبل است و حوصله ی تایپ کردن کامل نام رستوران را ندارد و اگر چند رستوران با آن پیشوند وجود داشت از نظر الفبایی اولین رستوران را انتخاب کند) و از آنجایی که او وسواس در انتخاب رستوران ها دارد قرار است به صورت پیوسته مقدار  $k$  و  $S$  را عوض میکند تا به نتیجه ی مطلوب خود برسد.

با این که امیر با ساختمان داده ها آشنایی دارد اما از آنجایی که گرسنه است مغز او به او فرمان نمی دهد تا رستوران مورد نظر خود را پیدا کند و از شما که دوست خوب او هستید میخواهد تا رستورانش را پیدا کنید. تنها کمکی که امیر به شما می تواند کند این است که ابتدا تعداد رستوران ها رو به شما بدهد سپس در  $n$  خط بعدی نام و سپس فاصله ی رستوران را از خودش را به شما بدهد.

سپس  $q$  که تعداد کوئری هاست را به شما می دهد.

در  $q$  خط بعدی در هر خط  $k$  و  $S$  داده می شود.

شما باید در خروجی  $q$  خط چاپ کنید که خط  $i$  ام نشانگر نام رستوران مورد نظر امیر در کوئری  $i$  ام است.

پیچیدگی زمانی:

در ابتدا شما باید از ساختمان داده ای استفاده کنید که پیدا کردن  $k$  رستوران نزدیک را در  $k * \log(n)$

انجام دهد. (راهنمایی: ساخت این ساختمان داده پیچیدگی زمانی  $O(n)$  را می طلبد)

از آنجایی که هر کوئری امیر وابسته به کوئری قبلی او می باشد شما لازم نیست که در هر کوئری برای پیدا کردن  $k$  رستوران نزدیک  $k * \log(n)$  هزینه کنید زیرا ممکن است تا قبل از این کوئری  $m$  تا رستوران نزدیک را داشته باشید و صرفا اگر  $k > m$  باشد  $(k-m) * \log(n)$  هزینه برای پیدا کردن  $k-m$  رستوران دیگر هزینه کنید. (زیرا  $m$  رستوران اول را قبلا بدست آورده اید)

حال باید نام این  $k$  رستوران نزدیک را داخل ساختمان داده ی دیگری ذخیره کنید تا هنگامی که  $s$  را امیر به عنوان ورودی کوئری می دهد بتوانید در  $k$  رستوران نزدیک ، اگر بلند ترین طول نام رستوران ها  $\text{max\_len}$  باشد با پیچیدگی زمانی  $O(\text{max\_len})$  رستوران مورد نظر امیر را پیدا کنید. (راهنمایی: از ساختمان داده ی درخت تری (Trie tree) استفاده کنید) توجه کنید که هم ساختمان داده ای که برای قسمت بالاتر در نظر گرفتید و همینطور این ساختمان داده هر دو باید پویا باشند یعنی با هر بار کوئری گرفتن تغییرات به صورت آنلاین روی این ساختمان داده ها انجام شود.

در صورتی که در کوئری  $i$  ام مقدار  $k$  را امیر بدهد و شما  $m$  رستوران نزدیک را داخل درخت Trie خود داشته باشید باید بتوانید با پیچیدگی زمانی  $O(|m - k| * \text{max\_len})$  (اگر  $m > k$ ) تعداد  $m-k$  رستوران دورتر از ساختمان داده ای که برای نام ذخیره نام رستوران ها ساخته بودید حذف کنید. و همینطور (اگر  $k > m$ ) تعداد  $k - m$  رستوران دورتر را به درخت اضافه کنید.

example

**Input:**

7

bluebird 5

bluelion 100

siboos 19

siniour 40

narenj 29

narenji 35

delta 77

5

3 blue

2 sib

1 sib

4 narenji

3 narenji

**Output:**

bluebird

siboos

NO restaurant found!

narenji

NO restaurant found!

