به نام خداوند جان و خرد

نام و نام خانوادگی:محمد حسین کریمی

شماره دانشجویی:9731833

پروژه جیمز باند

بهار1400

الگوریتم کلی

ابتدا جمله را بصورت یک نود در یک صف می ریزیم و چون ساختار صف FIFOهست از سمت چپ شروع می کنیم و کاراکتر به کاراکتر پیش می رویم

توضیح کد

این پروژه از نوع جاوا هست و از gradleداره استفاده می کنه چون از gsonاستفاده می کنیم برای دیکود کردن فایل jsonچون خود بطور پیش فرض نمیتونه json را دیکود کند.

متد getwords

این متد سعی می کند از ادرس فایلی که بهش دادیم میاد دیکشنری json را می گیره و با استفاده از jsonobject.keysetتمام keyهارو بر می گردونه و در یک setذخیره می کنه همان طور که می دانید setهیچ مقدار duplicateتوش وجود ندارد پس از این طریق مطمئن می شویم این setکه داریم دیکشنری کاملی هست که هیچ کلمه تکراری در ان وجود ندارد .همان طور که می دانید در فایل jsonهمه کلمه ها در قسمت keyقرار دارند و valueهمه انها برابر 1 است

Class node

یک کلاس کمکی است که دو مقدار val,parsedکه اگر یک مقدار بهش پاس بدیم در valذخیره میشه و اگر دو مقدار داشته باشیم در valوparsedذخیره می شود در ادامه به بررسی جزئیات ان می پردازیم.

متد word break

این متد اصلی برنامه است.ابتدا یک queueمی سازیم از جنس nodeکه چند تا nodeمی خوایم در ان قرار دهیم. و اولین چیزی که می خواهیم در ان بریزیم اون رشته ای است که به عنوان ورودی گرفتیم داخل queueمی ریزیم سپس یک حلقه می زنیم تا زمانی که صف خالی نشده.سپس اولین المنت را از صف بر می داریم که در واقع همان رشته ورودی ما است و ان را در نود قرار می دهیم که مقدار val ان رشته ورودی ما است سپس میایم چک می کنیم مقدار valان صفر است یا نه اگر صفر نبود یک حلقه دیگر بر روی کلمات دیکشنری می زنیم و چک می کنیم ایا مقدار نود val ما با کلمه ای تو دیکشنری ممکن شروع بشه یا نه.اگر کلمه ای پیدا کردیم این کلمه را در parsed قبلی که خالی است به علاوه یک space جمع می کنیم و دوباره این مقدار را بر می گردانیم تو queue.دوباره به forاول بازگشته و چک می کنیم مقدار valصفر شده یا نه این کار را تا زمانی انجام می دهیم که صف خالی شود.

پیچیدگی زمانی

O(n^n)

از آنجا که این مسئله با روش بازگشت به عقب حل شده است پیچیدگی زمانی آن با روش مونت کارلو بدست می آید. برای مثال کلمه ی فرضی زیر را درنظر بگیریم:

Abcde...

1+تعداد کل کلماتی که با حرف A شروع می شوند + تعداد کل کلماتی که با حرف A شروع می شوند (تعداد کل فرزندان این سطح)*تعداد کل کلماتی که با Ab شروع می شوند(تعداد فرزندان امیدبخش در این سطح)+تعداد کلماتی که با Ab شروع می شوند* تعداد کلماتی که با Abc شروع می شوند+...

اجرای برنامه

```
12:50:04 AM: Executing task 'WordBreak.main()'...

> Task :compileJava UP-TO-DATE
> Task :processResources NO-SOURCE
> Task :classes UP-TO-DATE
SOMESILENCESHOULDBETRANSLATED
> Task :WordBreak.main()
 some silence should be translated

BUILD SUCCESSFUL in 32s
2 actionable tasks: 1 executed, 1 up-to-date
12:50:37 AM: Task execution finished 'WordBreak.main()'.
```

```
12:17:02 PM: Executing task 'WordBreak.main()'...

> Task :compileJava UP-TO-DATE
> Task :processResources NO-SOURCE
> Task :classes UP-TO-DATE
IWANTTOGOHOME

> Task :WordBreak.main()
 i want togo home

BUILD SUCCESSFUL in 11s
2 actionable tasks: 1 executed, 1 up-to-date
12:17:14 PM: Task execution finished 'WordBreak.main()'.
```