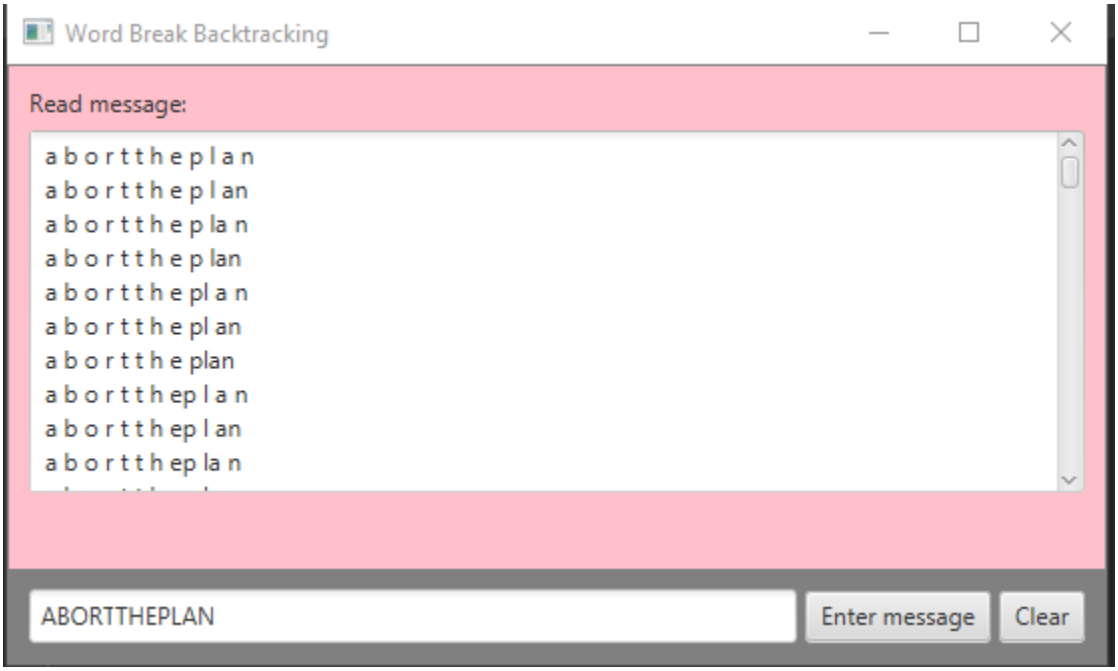
بنام خدا

گزارش پروژه ی درس طراحی الگوریتم (روش عقبگرد)

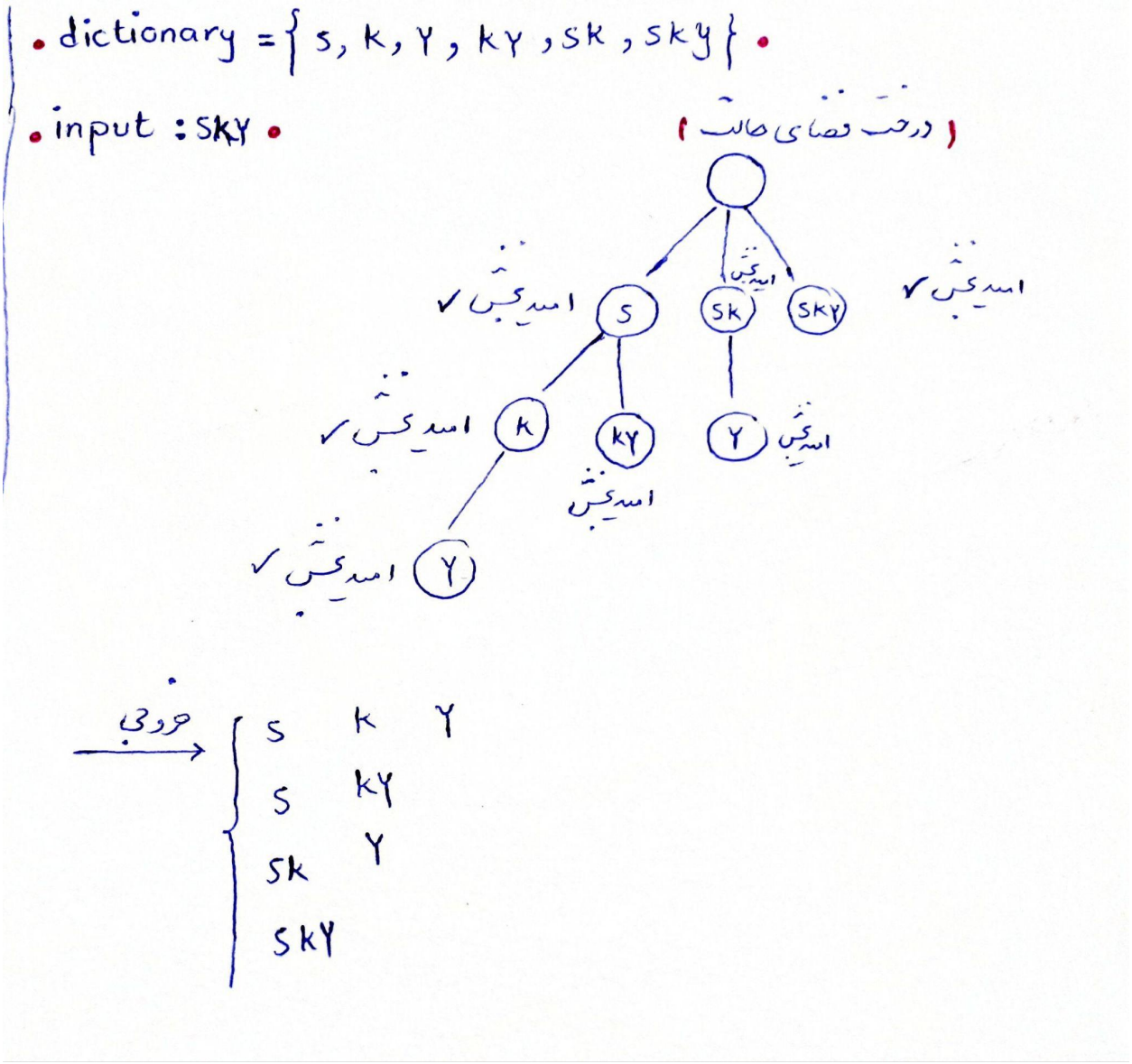
نمونه ای از اجرای برنامه :



روند اجرای برنامه و تحلیل پیچیدگی زمانی :

ابتدا همه ی کاراکتر های رشته ی ورودی را به حروف کوچک تبدیل می کنیم، و با استفاده از یک تابع با عنوان wordBreak تمامی جملات استخراج شده از ورودی را چاپ می کنیم برای این کار از روش عقبگرد استفاده می کنیم به این صورت که با استفاده از یک شمارنده تمامی پیشوند های رشته ی ورودی را بدست می آوریم اگر این رشته را یک گره در درخت فضای حالت فرض کنیم در نتیجه یک گره در صورتی امیدبخش است که در دیکشنری مورد استفاده ی ما موجود باشد، در بدترین حالت همه ی کاراکتر های رشته ی ورودی در دیکشنری موجود هستند، د درنتیجه اگر سایز رشته ی ورودی n باشد پیچیدگی زمانی از مرتبه ی n^n خواهد بود در نتیجه هر چقدر سایز دیکشنری کمتر باشد و حروف کمتری را پوشش دهد سرعت بیشتری خواهیم داشت.
برای نمونه برای ورودی SKY درخت فضای حالت بصورت زیر می باشد، در هر مرحله چک می شود که پیشوند رشته ی ورودی در دیکشنری موجود است یا خیر در صورتی که موجود باشد چک می شود که آیا این پیشوند کل رشته ی ورودی را پوشش می دهد یا خیر که در صورتی که پوشش بدهد خود

این رشته بعنوان یکی از جواب های ماست و اگر پوشش ندهد این رشته را به رشته ی نتیجه اضافه می کنیم و روی باقیمانده ی رشته تابع را فراخوانی می کنیم.



اگر از الگوریتم مونت کارلو برای محاسبه ی پیچیدگی زمانی کمک بگیریم و یک مسیر ویژه را انتخاب کنیم داریم :

$1 + n + n(n - 1) + n(n - 1)(n - 2) + \dots + n(n - 1)(n - 2)...1$

(فرض شده است که دیکشنری مورد استفاده بزرگترین دیکشنری ممکن است.)

همانطور که بدست آمده است پیچیدگی زمانی در بدترین حالت n^n است که به دلیل هرس شدن درخت فضای حالت در روش عقبگرد پیچیدگی زمانی بهتری دارد.