Assignment 1: Beam Search

The goal of the assignment is to implement beam search algorithm. You need to extract a directed weighted graph from a corpus and perform beam search on this graph to generate the sentence with highest probability.

**Task 1: Extract graph from a set of sentences.**

In this task, you will work on ExtractGraph.java or ExtractGraph.py.

The sentence dataset locates in “assign1\_sentences.txt”. Each line of file is one sentence, starting with “<s>”, and end with “</s>”. Punctuations include only “,” and “.”. It is easy to obtain the words simply by splitting the sentence with white space. Please keep the original lowercase and uppercase.

Your codes will extract a directed weighted graph from this dataset in the **ExtractGraph initialization step**. Each node represents a word; each edge connecting a head word and a tail word means the tail word is the next word of the head word; the edge weight is the probability of next word appearing after head word.

Also, you need to implement **getProb()**, which can read the graph and return the probability of next word appearing after head word.

**Task 2: Implement Beam Search on the graph to generate the sentences of max probability.**

In this task, you will work on BeamSearch.java or BeamSearch.py.

You need to implement **beamSearch()**.

**Pre\_words** is the existing words in the sentence, and you will need to predict next and following words to finish the sentence.

**beamK** is width of beam.

**maxToken** is the maximum words of a valid sentence, including the pre\_words.

Searched/generated sentence with its probability should be returned in form of StringDouble.

Search termination:

1. “<\s>” appears in the generated sentence. <\s> is the end of a sentence.
2. The count of words in the sentence is bigger than maxToken.

Existing online resources of Beam Search: https://geekyisawesome.blogspot.com/2016/10/using-beam-search-to-generate-most.html

**Requirements and Reminders:**

1. You **CANNOT** change the classes’ names or the required methods’ names. However, you can add new variables, constants, and methods in these classes and create new classes if necessary.
2. **Assignment1Main** is the main class for running your assignment 1. You are **NOT** allowed to change anything in this file.
3. You **CAN ONLY** use Java or Python in this assignment.
4. You **CANNOT** use external Java or Python packages.

**Grading:**

Your submission will be graded based on:

1. Correctness of the implementation on the required functions (70%)
2. Efficiency of your implementation, make sure your code finish processing two collections within 10 minutes (20%)
3. Necessary program annotation and commentaries (10%)

## **Submission Requirements**

A zipped file package with the naming convention as “pittids\_a1”. For example, suppose the Pitt id is jud1, then the submission package should be jud1\_a1.zip.

The file package should contain:

1. All the scripts/programs you used for this assignment (**src folder**)
2. Your output in the screen. (This should in **txt file**.)

**Do not upload the assign1\_sentences.txt.**