Assignment-1

- You are given a vocabulary of L words, V = {w_1,...,w_L}. Additionally, there are two special tokens <SoS> and <EoS>, that denote start of sentence (the default first word in any sentence, or w_0) and end of sentence (the default last word in any sentence, or w_{L+1}) respectively. You are also given a transition matrix P that contains (positive) scores for P(w_a|w_b) for all possible pairs of words. This score denotes the chances of seeing a word w_a given a word w_b.

- Based on this information, you need to generate a sentence of 'n+2' words including <SoS> and <EoS>. Ideally, we are looking for the sentence for which the following score is maximized:

$$S\left(w_0, w_1, \ldots, w_n, w_{L+1}\right) = P\left(w_{L+1}|w_n\right)\left(\prod_{j=1}^{n} P\left(w_j|w_{j-1}\right)\right) P\left(w_0\right)$$

  Here, each w_i for i=1,...,n is an element of V.

- The input to the code will be the numbers 'L' and 'n+2', a file containing the transition matrix (see sample file for L=4) and a file containing the vocabulary of L words (see sample file).

- The first L rows of the transition matrix file correspond to the transition scores for words {w_1,...,w_L} in the vocabulary given a particular word. E.g., the i^{th} row of this file contains the scores P(w_k|w_i) for k = 1,...,L. The $(L+1)^{th}$ row contains L scores P(w_i|w_0) for i = 1,...,L. The $(L+2)^{th}$ row contains L scores P(w_{L+1}|w_k) for k = 1,...,L. P(w_0) may be assumed to be 1.

- The output of your code should be a sentence of 'n+2' words including <SoS> and <EoS>. Also print its score 'S' in the next line.

- Example input:
  4 6 transition.txt vocab.txt

Tasks:

- Implement the following search algorithms for the above problem:
  (1) IDDFS
  (2) UCS

(3) Greedy search using an appropriate heuristic function of your choice
(4) A* search using an admissible heuristic function

- In the report, explain your intuition behind your approach in each case. Also explain and validate your algorithm using simple working examples.

- Modify the default IDDFS algorithm so that it returns an optimal solution for the above problem; i.e., a sentence of a given length with the maximum possible score 'S'.

- Run each of the above four algorithms at least 5 times for different values of n in {3,4,5,6} by randomly generating a transition score file each time, for different values of L in {3,5,10,15}. Analyze the average number of nodes explored by each algorithm for each value of n and L on an average, and the compute time. Also prove the admissibility of the heuristic function you use in A* search in your report.

Instructions:

(1) Prepare a single code file.
(2) Prepare a detailed report in PDF discussing all the steps, analyses, design choices and reasoning behind them. The PDF should be searchable and should not contain any code snippets.
(3) Modularize the code for readability wherever possible. Submit a single zip file containing your code, report [.pdf] and a readme [.txt] in google classroom under "Assignment-1". Name your files as <YourRollNumber_1.c> (if you write a code in c, otherwise replace this with an appropriate extension), <YourRollNumber_1.pdf>, <YourRollNumber_1.txt> and <YourRollNumber_1.zip>.
(4) A submission which does follow any of the submission-related guidelines will be awarded a penalty of 25% in this assignment.
(5) Confirmed cases of plagiarism will result in a zero in this assignment and an additional penalty in the total score in the course. Further, this submission will be considered in the top-2 submissions for grading.
(6) Strictly follow the Academic Code of Honor as given below, otherwise it will attract a penalty same as in point (5).
(7) There will be a penalty of 25% per day for late submissions. A submission which is >3 days late will not be evaluated. The time recorded in google-classroom will be considered.
(8) If you find any inconsistency in the problem description, discuss it as a public comment in google classroom under this assignment.

Academic Code of Honor:
(Adapted from https://stanford-cs329s.github.io/syllabus.html, Lecture-1)

(1) OK to search, ask in public about the topics we're studying. Cite all the resources you reference. E.g. if you read it in a paper, cite it. If you ask on Quora, include the link.
(2) OK to discuss questions with classmates. Disclose your discussion partners.
(3) OK to use existing solutions as part of your assignment. Clarify your contributions.
(4) NOT OK to ask someone to do assignment for you.
(5) NOT OK to copy partial/complete solutions from classmates or any other source.
(6) NOT OK to pretend that someone's solution is yours.
(7) NOT OK to post your assignment solutions online.
(8) If unsure, ask in google classroom as a public post under this assignment.