Deterministic models and optimization

Programming projects

For each project the tasks to complete are:

1. Write a text of at most 6 pages containing a discussion of the problem, the proposed algorithm, and a proof of correctness. The exposition should be clear, readable and well organized. Writing style and proper English usage will be taken into account for grading. The text should include all the references used in the project.

2. Write complete computer code of the algorithm, preferably in Python, R or C. Each line of the code must be commented indicating its meaning in the flow of the algorithm. Check correctness of your algorithm on small examples.

3. Report the output of the algorithm on the data sets provided in the statement of the project.

4. The project must be submitted as a **single** PDF file containing:

   (a) Discussion of the problem and proposed algorithm. Must include: proof of correctness; analysis of complexity as a function of the input size; a brief discussion of the main algorithmic paradigm use in the solution: greedy, divide and conquer, dynamic programming. . .

   (b) References used in the project.

   (c) Complete computer code with comments.

   (d) Output of the algorithm on the data sets.

   Submissions not following this format will not be considered.

Due: November 17

1 Edit distance

Given two strings of text X and Y, there are many ways to measure how much X and Y differ. Consider the following three operations on a string:

- D: Deletion of a character.
- I: Insertion of a character.
- S: Substitution of a character.

The edit distance $d(X, Y)$ is the minimum number of operations $\{D, I, S\}$ needed to perform on X to produce Y.

Task.

1. Design an algorithm that, given strings X and Y, computes the edit distance between X and Y. The algorithm should provide also the optimal sequence of operations transforming X into Y.
2. Modify the previous algorithm with a penalty function: operations D and I have unit cost 2, whereas operation S has unit cost 1.

Data. Run both algorithms on the following pairs of input strings and report the edit distance

1. (DNA)

   X = ACTACTAGATTACTTACGGATCAGGTACTTTAGAGGCTTGCAACCA

   Y = TACTAGCTTACTTACCCATCAGGTTTTAGAGATGGCAACCA

2. (Proteins)

   X = AASRPRSGVPAQSDSDPCQNLAATPIPSRPPSSQSCQKCRADARQGRWGP

   Y = SGAPGQRGEPGPQGHAGAPGPPGPPGSDG

2 Huffman codes

Given an alphabet A, a *code* replaces each letter x of A by a variable-length binary string $c(x)$. A code is a *prefix code* if for distinct letters x and y in A, the string $c(x)$ is not a prefix of $c(y)$. A prefix code can be decoded unambiguously scanning the encoded message from left to right.

Given a text T, let f_x be the frequency of letter x in T. The average number of bits required per letter is the quantity

$$C = \sum_{x \in A} f_x |c(x)|,$$

where $|c(x)|$ is the length of the string $c(x)$. A prefix code is optimal if C is minimal among all prefix codes.

Task.

1. Design an algorithm that, given an input text T, constructs an optimal prefix code for T. The *size* of the input is the number of characters in T.

2. Design an algorithm that, given a prefix code for a text T, outputs T.

Data. Run your algorithm on the following text to produce an optimal prefix code. Blanks, dots, questions marks, etc. are part of the alphabet. Upper and lower cases are considered the same letter. Write explicitly as a table the encoding function $c(x)$.

Text 1. O all you host of heaven! O earth! What else? And shall I couple hell? Oh, fie! Hold, hold, my heart, And you, my sinews, grow not instant old, But bear me stiffly up. Remember thee! Ay, thou poor ghost, whiles memory holds a seat In this distracted globe. Remember thee! Yea, from the table of my memory I'll wipe away all trivial fond records, All saws of books, all forms, all pressures past That youth and observation copied there, And thy commandment all alone shall live Within the book and volume of my brain, Unmixed with baser matter. Yes, by heaven! O most pernicious woman! O villain, villain, smiling, damned villain! My tables! Meet it is I set it down That one may smile, and smile, and be a villain. At least I'm sure it may be so in Denmark. So, uncle, there you are. Now to my word.

Text 2. Habe nun, ach! Philosophie, Juristerei und Medizin, Und leider auch Theologie Durchaus studiert, mit heissem Bemühn. Da steh ich nun, ich armer Tor! Und bin so klug als wie zuvor; Heisse Magister, heisse Doktor gar Und ziehe schon an die zehen Jahr Herauf, herab und quer und krumm Meine Schüler an der Nase herum Und sehe, dass wir nichts wissen können! Das will mir schier das Herz verbrennen. Zwar bin ich gescheiter als all die Laffen, Doktoren, Magister, Schreiber und Pfaffen; Mich plagen keine Skrupel noch Zweifel, Fürchte mich weder vor Hölle noch Teufel Dafür ist mir auch alle Freud entrissen, Bilde mir nicht ein, was Rechts zu wissen, Bilde mir nicht ein, ich könnte was lehren, Die Menschen zu bessern und zu bekehren. Auch hab ich weder Gut noch Geld, Noch Ehr und Herrlichkeit der Welt; Es möchte kein Hund so länger leben! Drum hab ich mich der Magie ergeben, Ob mir durch Geistes Kraft und Mund Nicht manch Geheimnis würde kund; Dass ich nicht mehr mit saurem Schweiss Zu sagen brauche, was ich nicht weiss; Dass ich erkenne, was die Welt Im Innersten zusammenhält, Schau alle Wirkenskraft und Samen, Und tu nicht mehr in Worten kramen.

Comment briefly on the differences between the codes associated to the two texts.