A Tentative Approach to Find Acrostics in Latin Texts

Feiyang Liu

March 2020

1 Introduction

Acrostic is a type of practice to form a "meaningful structure" by the first letter of subsequent lines of the poem. [1] Commentators throughout the years tried to find acrostics and interpreted their appearances in different Latin texts. With the resource of a plain txt file of *A Latin Dictionary* (edd. Charlton T. Lewis and Charles Short) shared by `Telemachus` from github, I have made a program that can shorten the searching processes of acrostics and provide as the output a list of words that are potentially made acrostics by Latin authors. Further validations on their appearances require the readers to look back into the actual verses of Latin texts and interpret their nearby circumstances with allusions or any other literary devices.

2 Internal Design

Before the commencement of the project, a general guess on what kind of words that can be acrostics is crucial. Unfortunately, I find all those words that have inflections can be represented as good candidates of acrostics. Therefore, an acrostic can be a word of nouns (pronouns are hard to verified as acrostics), verbs, adjectives (some adverbs can be transformed from adjectives), or interjections (not inflection, but possibly a strong signal to draw reader's attention). Furthermore, an acrostic does not necessarily have to be in the standard form (I define the **standard form** of a word to be the first word in each dictionary entry, i.e., nominative form for nouns or first person singular present indicative active for verbs), for example, "*VNDIS*" in *Eclogues* 9.34-38. [2] As currently I do not have the ability to let the program identify the grammatical forms of the word, I pick the standard form of each word (each word with its explanation takes one line in the txt file of *Lewis and Short*) and compare the possible acrostic with them. In this case, for simplicity, we only deal with normal acrostics (consecutive lines) and alternate-line acrostics.

2.1 Searching Processes and Factors

There are three steps to form the lists of acrostics:

(a.) Form a list of words with word length ≥ 2 (one letter does not represent anything) in the standard form from *Lewis and Short*

(b.) Form a string that consists of all initial letters of verses by sequence from Latin texts.

(c.) Form a substring with certain word length for every index and check whether the dictionary list has a similar form.

For step (a.), since *Lewis and Short* has words with beginning letter, "j" and "v", we have to consider when we should convert them to their corresponding vowel forms, "i" and "u", because not every Latin text will specify consonant forms of "i" and "u". When we should convert upper case of letter to lower case is

[1] Richard F. Thomas and Jan M. Ziolkowski, eds., *The Virgil Encyclopedia*. 3 vols. Wiley-Blackwell, Malden MA 2014, p. 8

[2] Richard F. Thomas and Jan M. Ziolkowski, eds., *The Virgil Encyclopedia*. 3 vols. Wiley-Blackwell, Malden MA 2014, p. 8

another problem. A feasible method is to exclude all words started with capital letters from the dictionary list, but the output will omit the famous "*MARS*" in *Aeneid* 7.601-4. For these two problems, we decide to be inertial and convert them only when comparison is necessary in part (c.). Furthermore, every dictionary entry will have marked *macron* or *breve* to indicate whether the vowel is long or short and we have to convert them back to the normal form. Finally, in order to extract the first word of the entry, we will stop taking characters from every line of the txt file as long as the character is not a letter (its ASCII code is not within certain range).

For step (b.), we simply take the first letter of each line from Latin texts as long as the line contains a letter. Originally, we should be careful for different kinds of punctuation marks and empty spaces, but it can be solved by looking at the range of ASCII code.

For step (c.), if we scrutinize the strings formed in step (b.), we will discover that there are hundreds of consecutive consonants, but only a few of the consonants can appear in certain orders in Latin words. Therefore, we decide to use a map to store the dictionary list, where the key of the map is the first two letters of a Latin word and the value of the map is a list of Latin words that have the first two letters as the key. Hence, we can see a brute-force algorithm in the next section.

2.2 Algorithm

```
Create an empty map.
for each Latin word in the standard form from dictionary list:
    if (map does not contain a key as the first two letters of the word):
        Put the first two letters of that word into the map as a new key.
    Add the word to the corresponding list of that key.

for i from 10 to 4:
    for each index j of the string s formed in step (b.):
        // substring s1 is a portion of the acrostic string from the Latin text.
        Create a substring s1 = s.substring(j, j + i)
        if (s1.substring(0, 2) is a key of the map):
            Find the first word in the map that has the first
              (floor(s1.length()) / 2 + 2) letters same as that of the substring s1.
            Return that word.
```

Two decisions are arbitrary: (1). the range of "i" is from 4 to 10, (2). we only compare the first $(floor(s1.length())/2 + 2)$ letters between the substring and the dictionary word (and only return the first one that satisfies).

For the first decision, a word can be too short to be identified as an acrostic, so we have to compare the word that has at least length 4. Furthermore, a word with length 10 is currently long enough for acrostic except rare conditions because the poet should be so talent that he can pick words with certain initial letters meanwhile the word sufficiently satisfies the meter, literary style, and circumstances of the nearby lines for more than 10 lines.

For the second decision, since the Latin word is inflected with endings and a word is more feasible to be an acrostic when it is longer, we decide to use this formula to decide how many letters of the substring that we want to compare: $(floor(s1.length())/2 + 2)$. For example, if a substring has a length 4, then we have to compare the first 4 letters of the substring (which is the entire substring) with other dictionary words (we can regard that first 4 letters as an imaginary stem of some Latin words); however, if a substring has length 5, we only compare its first 4 letters with the first 4 letters of other dictionary words (and there will be more leniency as the substring is longer).

Finally, for efficiency, we only return the first word instead of the best word that satisfies the above conditions. At the current stage, it is hard to define criteria for "the best," which bases on the reader's acquaintance of the word, first letter capitalized, usage, etc. Therefore, we pick the most approachable choice, the first appearance. In addition, if the map has a list of n elements, finding the best element

satisfying certain conditions requires to loop through all elements, which is $O(n)$, but finding the first element satisfying certain conditions requires, for the best case, $O(1)$ run-time.

Hence, unfortunately, by the above decisions, we cannot find "*ALMA*" with alternate lines in *Aeneid* 1.1-7 [3] because the substring has a length 4 and the corresponding word in the standard form is "*ALMUS*", which has a stem with length 3. At the same time, if we let loose the criteria for choosing potential acrostics (i.e., change the formula in the second decision), stream "*ALMO*" will be returned by the algorithm before looking at "*ALMUS*".

3 Manual

3.1 Download Files

The program is made as an open source online shared in github. This is the link to get all sources that you need to run the acrostic program:

`https://github.com/liuf9/hiddenword`

where you can find and download the executable jar, `hiddenword.jar` by its name, under

`out/artifacts/hiddenword_jar`

You also need a dictionary text, which includes all Latin words with word length ≥ 2 in "*Lewis and Short*". You can find and download this text file, `dict.txt` by its name, under

`src/com/latin`

(Whenever you find it hard to download the dictionary text, click the file to see its content. When you find the top edge of the file has some words as "51520 lines", read the entire line and find `"Raw"`. Then click it and you will find a new page that shows the content of the file. Right-click and save the file as `"Page Source"` in the format with the name `dict.txt`.)

I include some of the output text files in the same directory as `dict.txt`, which all are collected by the name of the Latin text.

3.2 Execute the Program

(ω.) Check whether your computer has `Terminal` (if you are using Windows, you might need to install `"cygwin"`, but I personally do not want you to install new softwares) and has Java (open a window of `Terminal`, which you can find under `"Applications"` folder and type command `"java -version"` (and press `"Enter"`). If the terminal does not normally show you java version, you might want to install `"Java"`). Therefore, if you do not have the above two conditions, I do not recommend that you do the following steps.

(a.) Open a window of `Terminal` at the folder (I would recommend you to create a new folder for simplicity in the next few steps) where you have saved your executable jar (if you are using Mac, you can simply right-click that folder and click `"Services"` and then click `"New Terminal at Folder"`.)

(b.) (Optional) Use command `"ls"` (which stands for "list") to check whether the executable file is in the folder. (Namely, for the next few steps, always type your command in `Terminal` and press `"Enter"`.)

(c.) Type `"java -jar hiddenword.jar"` (or copy the command in the quotes) and press `"Enter"`. If the terminal asks you `"What is the dictionary address (no space):  "`, that means you are running the program now, feel free to use `"Ctrl+C"` to quit.

[3] Robinson, M. (2019). LOOKING EDGEWAYS. PURSUING ACROSTICS IN OVID AND VIRGIL. *The Classical Quarterly*, 69(1), 290-308. doi:10.1017/S0009838819000375, p. 299

(d.) Now you have to tell the address of the dictionary file. Mainly, you follow the process as part (a.) to open a new window of `Terminal` at the folder where you have saved your dictionary txt file and use command `"pwd"` (which stands for "print work directory") to print your address of the dictionary file. Copy (I strongly recommend that you COPY rather than TYPE) the address to the first Terminal window and press `"Enter"`.

(e.) The terminal will ask you "input address" and "output address" in the next two steps, so do the same process as part [(d.)] to find the addresses of the folders where you have saved your Latin texts and where you want to save your acrostic output files.

(f.) Then the terminal will ask you the name of the Latin text and the number of books it has, follow its instruction (your answer should not include any space, or it may crash). (In this case, make sure the books of Latin text is named in exactly this style: `"book" + number + ".txt"` (e.g., `""book1.txt"`) and if you answer 10 books, you must include 10 txt files that are named in this format, otherwise, the program will crash. If you do want to only find the acrostic word in Book 7, rename it to `"book1.txt"` and tell the program the text has "1" book.)

(f2.) Make sure your txt file includes only the clean Latin text, which means there is no title or any other irrelevant sentences in the file. The file is allowed to have empty lines and line numbers attached at the end of the line as we only look at the first letter of the line.

(g.) Next, the terminal will ask you whether your Latin text includes any letter, "v" or "j", both of which are the consonant forms of the vowels, "u" and "i". If you are not sure, for the worst case with caution, answer `"yes"`.

(h.) Finally, the terminal will ask you whether the Latin text written in Elegiac Couplet. This is a question that really asks you whether you want to find acrostics by alternate lines, odd-number lines or even-number lines. If you only want to find acrostics by consecutive lines, answer `"no"`; otherwise, the terminal further asks you to whether check hexameter verse (odd) or check pentameter verse (even) or normally check every line (consecutive), answer it with the letter, 'h' or 'p' or 'n'.

(i.) You should find the output files in a folder named by the name of the Latin text in the output address if everything works soundly.

3.3 Output File Usage

Every output file will have a list of words that might be a potential acrostic, but it will also include lots of false positives. The display of the output is ordered by the length of the substring in decreasing sequence (which is a portion of the entire acrostic string formed in Section 2.1, step (b.)). Every line represents a possible acrostic found in the Latin text, where in the first quote, there will be a dictionary entry, together with the number of characters in the given line. Generally, the number of characters in a line is approximately proportional to the length of the sentence and a greater number of characters implies a more detailed explanation of the word. In the second quote, there will be the printed substring which we used to compare in Section 2.2. Three indicators can help to judge whether the substring is an acrostic:

(a.) the number of characters used to explain the word (a lengthy description generally implies a common usage of the word in the text)

(b.) whether the first letter is capitalized (capitalized first letter generally implies a name for site, nation, or god, etc., and if it is content-irrelevant, it has a very low possibility to be the acrostic)

(c.) the length of the dictionary word (longer word with more correspondence between the word and the substring implies a greater possibility)

(d.) we can always check whether the substring in the second quote reminds us some familiar words.

We can take "*MARS*" in *Aeneid* 7.601-4 as an example:

```
Find word "Mars (5866)" in form "mars" from line 601 to line 604.
```

"*Mars*" is the word that has a lengthy description (with 5866 characters), which implies that it has a common use in classical Latin, and the capitalized first letter implies it is a name instead of a common noun. The two numbers indicates the range of lines that the acrostic is in.

4 Potential Acrostics in Poems

4.1 *Metamorphoses*: Exception for Usages

The criteria in the section 3.3 may not always work in finding potential acrostics. We can take the example given by Robinson in his paper, LOOKING EDGEWAYS. PURSUING ACROSTICS IN OVID AND VIRGIL. [4] We shall look at *Metamorphoses* 8.533-38 [5],

```
Find word "nitido (451)" in form "nitidopa" from line 531 to line 538.
Find word "nitide (29)" in form "nitidop" from line 531 to line 537.
```

Although the words are shown in the beginning lines of the output file, the number of characters does not show that the word is as popular as "*MARS*" in *Aeneid* 7.601-5 or "*DEUS*" in *Metamorphoses* 1.29-32, but the contextual references show a great inclination on the word as an acrostic. Therefore, we have to always go back to the Latin text and find confirmations on the words as acrostics in the final steps.

4.2 *Fasti*: Acrostics or Alternate-Line Acrostics

Normally, I take epics as the target poem style for acrostic test because dactylic hexameter does not change its meter between lines and the epic narrative is consistent. Then is Elegiac Couplet necessarily alternate-line acrostics? I take *Fasti*'s Book 1 [6] as a short example. The normal acrostic files has 45 outputs (see the website), in which the following line might show an acrostic:

```
Find word "hiems (3058)" in form "hiemsq" from line 593 to line 598.
```

(Line number might differ in different manuscripts. In this case, the actual line number might start from 595 to 600. The causation of this might come from the fact that some line numbers are flipped or having an ending letter "a", but the program does not detect this situation. So be aware of this if you want to use the program for Lucretius.) The acrostic is in the description of January 13th and 14th, which is of course a day in winter. However, within these lines, the poet expresses his laments on the achievements and early death of Nero Claudius Drusus Germanicus. Coincidentally, several authors argued that January 13th or 14th is the birthday of Nero Claudius Drusus Germanicus. [7] [8] Hence, "*HIEMS*" is a marker for one important date to Nero Claudius Drusus Germanicus.

Nevertheless, alternate-line acrostic files has 9 outputs for hexameter lines and 16 outputs for pentameter lines, both of which do not have a very strong candidate for acrostics. Hence, at the present stage, even if the meter is different from dactylic hexameter, we may still have a preference at acrostics by consecutive lines.

[4] Robinson, M. (2019). LOOKING EDGEWAYS. PURSUING ACROSTICS IN OVID AND VIRGIL. *The Classical Quarterly*, 69(1), 290-308. doi:10.1017/S0009838819000375

[5] The digital text for *Metamorphoses* is from "`The Latin Library`" (you can click the hyperlink here) and the digital text for *Aeneid* is from "`Gutenberg Project`".

[6] The digital text for *Fasti* is from "`The Latin Library`". From now on, without certain specifications, the digital Latin text is assumed to be taken from "`The Latin Library`".

[7] Barrett, Anthony A. (2002). *Livia: First Lady of Rome.* New Haven: Yale University Press., pp. 313-4

[8] Powell, Lindsay (2011). *Eager for Glory: The Untold Story of Drusus the Elder, Conqueror of Germania.* Barnsley, South Yorkshire: Pen & Sword Books., p. 3

4.3 First Letter of Line or First Letter of Word

If we do not consider the word order, sentence break-point, or other literary devices within the line, we will expect that the first letter of a line might have a similar distribution as the first letter of a word in general. Although some factors may influence the final distribution, they might not be as influential as the acrostics because whenever we want to form acrostics, we need to prepare a list of words started with vowels and put them at the beginning of the line. (A Latin word is hardly without a vowel and acrostic implies some lines must start with a vowel.) Hence, we take *Aeneid*, *Metamorphoses*, and *Pharsalia* as an example. For those three texts in Figure 1, 2, and 3 (the dotted line represents the average of a letter that can be the first letter of a line or a word and we omit letters "j", "v", "w", and "y"), the usage of "a", "e", "i", "o" in the first letter of a line is greater than that in the first letter of a word. Although this does not necessarily give us a conclusion of acrostic practices in the three texts, if you scrutinize and compare *Pharsalia* with the other two texts (worked as control groups), you will find a more drastic increase in the usage of vowels at the beginning of a line in *Pharsalia*. Therefore, it may suggest that we find more possibilities of acrostics in *Pharsalia*.

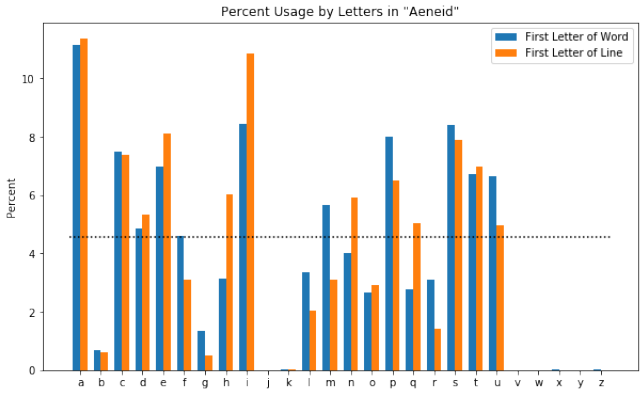

Figure 1: Letter Usage Comparison in "*Metamorphoses*"

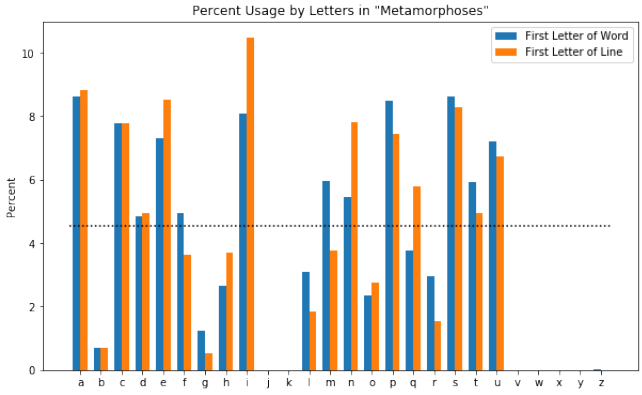

Figure 2: Letter Usage Comparison in "*Metamorphoses*"

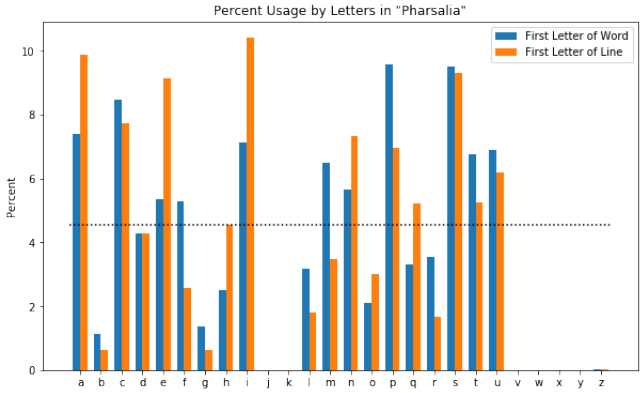

Figure 3: Letter Usage Comparison in "*Pharsalia*"

5 Conclusion and Improvements

At present, we have a decent list of output files that contain some candidates of the acrostics. We will not consider implementing algorithms for "*telestics*" (which consists of the last letter of lines) before a new optimization on searching algorithms for acrostics because for telestics, for the algorithm, we have to consider the influences of line numbers in the Latin text and for the poets, as Latin words end with vowels and specific consonants such as "s", "t", "m", etc. for the most of time, it is harder to form a telestic than an acrostic. For example, Robinson suggests a usage of telestic in *Metamorphoses* 1.452-5 [9], which is "*NASO*", the cognomen of Ovid. However, it is very hard to form a similar telestic for "*MARO*" as "r" is not a very common ending in Latin word.

If an algorithm to identify the grammatical form of the Latin word is available, we can both apply it to the substring and the Latin texts. For the latter, we can add more weights on the words that appear in the context and some words can then be added to the dictionary list such as "Iopas", the bard at the court of Dido.

A current improvement is to look at the list of acrostics left by Isidor Hilberg in his paper, *'Ist die Ilias Latina von einem Italicus verfasst oder einem Italicus gewidmet?'* [10], where he puts thousands of acrostics as accidental. From his decisions, we may find a pattern to abandon some clearly false positives from the output files.

References

[1] Richard F. Thomas and Jan M. Ziolkowski, eds., *The Virgil Encyclopedia*. 3 vols. Wiley-Blackwell, Malden MA 2014

[2] Robinson, M. (2019). LOOKING EDGEWAYS. PURSUING ACROSTICS IN OVID AND VIRGIL. *The Classical Quarterly*, 69(1), 290-308. doi:10.1017/S0009838819000375

[3] Barrett, Anthony A. (2002). *Livia: First Lady of Rome*. New Haven: Yale University Press.

[4] Powell, Lindsay (2011). *Eager for Glory: The Untold Story of Drusus the Elder, Conqueror of Germania*. Barnsley, South Yorkshire: Pen & Sword Books.

[5] I. Hilberg, *'Ist die Ilias Latina von einem Italicus verfasst oder einem Italicus gewidmet?'*, WSt 21 (1899), 264- 305

[9] Robinson, M. (2019). LOOKING EDGEWAYS. PURSUING ACROSTICS IN OVID AND VIRGIL. *The Classical Quarterly*, 69(1), 290-308. doi:10.1017/S0009838819000375, p. 293

[10] I. Hilberg, *'Ist die Ilias Latina von einem Italicus verfasst oder einem Italicus gewidmet?'*, WSt 21 (1899), 264- 305

A Appendix

In this section, we will select some examples that are potential candidates for the acrostics from different poems. Notice that with a slight change on the code, it is also able to identify some candidates for acrostics in English poems. We can take Milton's *Paradise Lost* Book 1 [11] as an example:

Let us not slip th' occasion, whether scorn,
Or satiate fury yield it from our Foe.
Seest thou yon dreary Plain, forlorn and wilde,
The seat of desolation, voyd of light,
Save what the glimmering of these livid flames
Casts pale and dreadful? ...

Milton, *Paradise Lost* 1.178-83

Peor his other Name, when he entic'd
Israel in Sittim on their march from Nile
To do him wanton rites, which cost them woe.
Yet thence his lustful Orgies he enlarg'd
Even to that Hill of scandal, by the Grove
Of Moloch homicide, lust hard by hate;
Till good Josiah drove them thence to Hell.

Milton, *Paradise Lost* 1.412-8

A.1 *Eclogues*

We begin with Vergil's *Eclogues* as every eclogue is not long and Vergil alternates narratives (e.g., dialogue or antiphony) in the odd indexed poems (which may decrease the appearance of acrostics).

M. ...
Nos patriam fugimus; tu, Tityre, lentus in umbra
Formosam resonare doces Amaryllida silvas.

T. **O** Meliboee, deus nobis haec otia fecit.
Namque erit ille mihi semper deus, illius aram
Saepe tener nostris ab ovilibus imbuet agnus.

Vergil, *Eclogues* 1.4-8

M. ...
Exiled from home am I; while, Tityrus, you
Sit careless in the shade, and, at your call,
"Fair Amaryllis" bid the woods resound.

T. O Meliboeus, 'twas a god vouchsafed
This ease to us, for him a god will I
Deem ever, and from my folds a tender lamb
Oft with its life-blood shall his altar stain.

[11] The digital text for *Paradise Lost* is from "Gutenberg Project". From now on, without certain specifications, the digital Latin text is assumed to be taken from "The Latin Library".

(tr. Greenough)

An acrostic, "*FONS*" meaning "a spring, fountain", appears at the beginning of *Eclogue* 1. "*FONS*" also has an implication of "source" or "origin", which can reflect back to this location, the beginning of this poem or this entire work. Furthermore, also notice that in line 39 and 52, an inflection of "*fons*" recurs in the poem. In the second appearance, Meliboeus says

> "Fortunate senex, hic inter flumina nota
> Et fontis sacros frigus captabis opacum."

Vergil, *Eclogues* 1.51-2

"Fortunate old man, here among the familiar rivers and sacred springs you will find the cool shade."

At the beginning of poem, Vergil creates a scene where both men recline on the grass of woods under the shades cast from the high trees which ease their minds from the "burning" fields and worries. By line 51-2, we know that shades and flowing waters always associate with each other. We might suggest that in the scene created by Vergil in line 1-3, a physical spring flows by their resting place, just as "*FONS*" passes from Meliboeus' words to Tityrus' words.

> *D.* ...
> **I**ncipe Maenalios mecum, mea tibia, versus.
> **N**unc scio, quid sit Amor: duris in cotibus illum
> **A**ut Tmaros, aut Rhodope, aut extremm Garamantes,
> **N**ec generis nostri puerum nec sanguinis edunt.
> **I**ncipe Maenalios mecum, mea tibia, versus.
> **S**aevus Amor docuit natorum sanguine matrem
> Commaculare manus; ...

Vergil, *Eclogues* 8.42-8

> *D.* ...
> "Begin, my flute, with me Maenalian lays.
> Now know I what Love is: 'mid savage rocks
> Tmaros or Rhodope brought forth the boy,
> Or Garamantes in earth's utmost bounds-
> No kin of ours, nor of our blood begot.
>
> "Begin, my flute, with me Maenalian lays.
> Fierce Love it was once steeled a mother's heart
> With her own offspring's blood her hands to imbrue: ...

(tr. Greenough)

An acrostic, "*INANIS*" meaning "empty, vain, void", appears in the second half of Damon's song in *Eclogue* 8. The song comes from a man who lost his love (as his loved one, Nysa, is married to another man, Mopsus) and at the end of the song, he prepares to commit suicide by jumping off a cliff. Notice that the four lines where Vergil begins his acrostic is a turning point from the previous parts, both in context and in line calculation. The previous stanzas before line 42 describe the facts and pasts (admittedly, including some emotional expressions, who will not be sorrowful when he/she tells the sad affairs), but after line 42, the song is largely composed of emotional expressions, more of desperation. Robinson suggests that the middle of the

poem is likely to find acrostics. [12] If we begin with line 21 where our protagonist formally starts his song (he begins with "*incipe*" at every stanza afterwards) and end with 61 (the final line begins with "*desine*", but the same format as before), the middle point is 41 and our acrostic starts at line 42, which is approximately the middle of the song. [13] (If we take the nine stanzas which all begin with "*incipe Maenalios mecum, mea tibia, versu*", the middle stanza (fifth) exactly shows the history of how our protagonist fell in love with Nysa and it reveals a symmetry in the song, for example, stanza 9 says farewell to the forests in stanza 1, stanza 2 describes some abnormal incidents and stanza 8 expresses a wish of more such incidents, etc.) In the modern usage of acrostics, acrostic usually represents the answer of a riddle formed by the corresponding lines. In this poem, our protagonist says, "*Nunc scio, quid sit Amor.*" "Then what is love?" you may ask. "*Amor est inanis.*"

[12]Robinson, M. (2019). LOOKING EDGEWAYS. PURSUING ACROSTICS IN OVID AND VIRGIL. *The Classical Quarterly*, 69(1), 290-308. doi:10.1017/S0009838819000375

[13]Personally, I am not a proponent of the idea which is to explicitly calculate the middle point of the poem, but we can always take it as a reference. Robinson discussed "*AUSUM*" in *Aeneid* 7.178-82, but such explicit calculation associated with the middle point of the poem implies that Vergil either accurately predicted how many lines he would write in the future (an astonishment for me if the poet only processed four lines in one day) or he made revisions on this part before his death. The second assumption further implies that Vergil did this revision before the incomplete lines, which might suggest that those lines are intentional.