**What were possible approaches considered?**

|  |
| --- |
| There were not many approaches I considered for this project since the instructions were straight-forward for what was required to make this project. I did, however, consider 2 different approaches that did not require using binary trees. |
| 1. I initially just figured using a dictionary would be the ideal approach. I’ve done an encoder/decoder program on python using dictionaries and figured that would be pretty similar to what was needed here (though it was not using Morse code and created code by adding various numbers to the character ascii). I still believe that would make for a much shorter code, but I think that approach would have not included binary trees at all which would defeat the purpose of this exercise. I did end up using one a bit, but I still use the binary tree |
| 2. The first thought I had for implementing this project was heaps. This may cause issues, however, since heaps seem to have a better usage trying to find the least/greatest for given characters/ints/etc than for just making the binary tree. It might have worked for this, but I don’t believe the order would have been as requested. |

**What were difficulties you encountered making the project?**

|  |
| --- |
| 1. One big difficulty was trying to figure out how to make the binary tree visible in a tree formation. Initially, I had no idea how to do this and could only print it out just as a list (Though I could get the height to work it seemed). It took me the longest to do that part and I actually ended up having to restart from scratch before I got it right |
| 2. Another problem was trying to get the code to decode correctly. I was overthinking this part severely so it became a lot more complicated that it should have been. There are error checks involved so I believe I’ve done this to the best of my abilities |

**Write down what you learned:**

|  |
| --- |
| While I’ve known the concept of binary trees, this was the first application of it in programming I’ve done. I’ve learned how efficient binary search trees can be and to hear alternatives to maps or vectors. I would have assumed that a bst would be less efficient in general. Doing this project has also helped provide more information on decoding/encoding works and different approaches you can take to doing them. |

**Feedback about the project (comments, suggestions for improvement, etc.)**

|  |
| --- |
| 1. I like the Morse code idea a lot. I think it’s a fun and creative project that helped me to better visualize binary trees a bit more. |
| 2. One improvement I would have liked would be a lot more comments for the binary search tree code that was included on canvas. It was a confusing trying to understand what all the pieces were doing, particularly since while I understand the concept of binary trees, I’ve never attempted to code them before. It was particularly frustrating since the code on canvas didn’t actually work without a lot of editing. |
| 3. For future classes, I think it would be helpful that the bst code on canvas be unlocked a bit earlier (at least for summer courses) to allow for testing/coding as much as possible before the due date, again with more comments. |