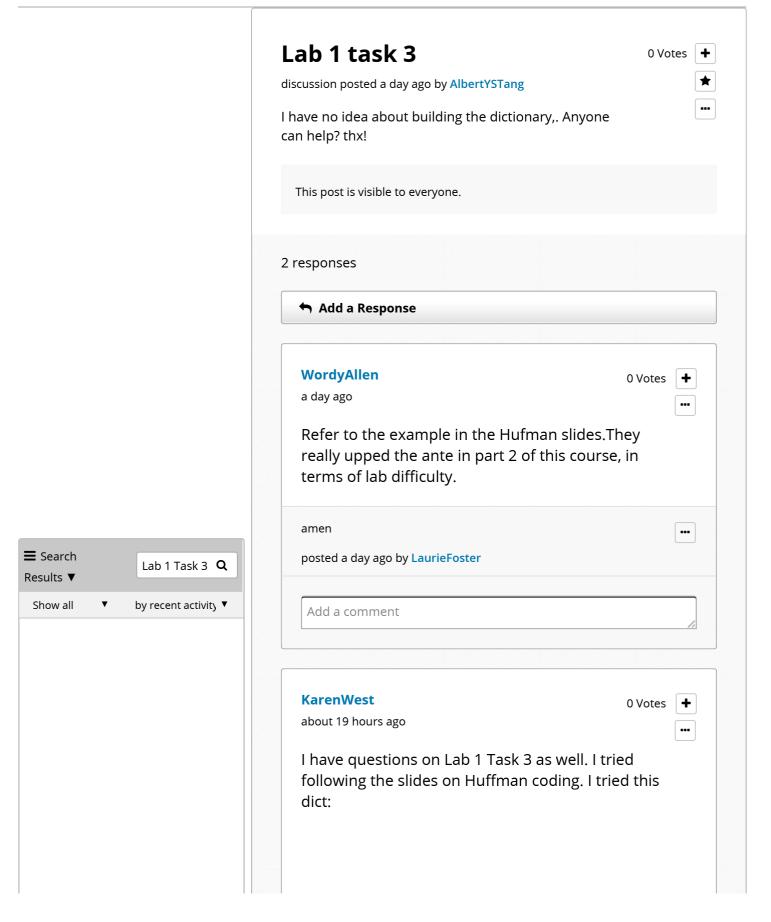


HKUSTx: ELEC1200.2x A System View of Communications: From Signals ...



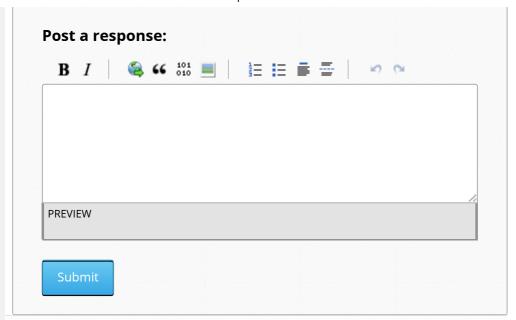
```
✓Zero run length?
                           3
? LAB 1 TASK 3
QLab 1 task 3
? Lab1, Task 4
```

```
Discussion - ELEC1200.2x | edX
     dict = \{[0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1], [1],
[0 1], [0 0 1], [0 0 0 1],...
     [0 0 0 0 0 1], [0 0 0 0 0 0 0 1],
[0 0 0 0 0 0 1], [0 0 0 0 0 0 0 0
1],...
     [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1], [0 \ 0 \ 0 \ 0
1]};
It came up with a Huffman Code Length of 121987,
```

so an improvement from the original code, but when I submitted it for grading, knowing it has to go down to 117374, it gave me an error and said the first dict vector is incorrect. I did this simply by adding a 0 for each increasing probability in the dict. I know in the slides, you added the least 2 probabilities to shorten the dict length for the next round of the algorithm until you reach 1. So my ordering of the probabilities by adding zeroes before reaching the number 1 did not take into account any adding of the 2 lowest probabilities (if it was supposed to!) I then tried just making it up, the most probably being 1, next probable 01, next 001, and then did the remaining vectors in 4 bits each, but it complained that the index exceeds matrix dimensions when I did it that way. I'm confused with this if anyone can help!

Maybe this will help you: https://en.wikipedia.org/wiki/Huffman_coding posted about 14 hours ago by machineinventing	•••
posted about 14 hours ago by machinemiventing	
Thanks - I found this link quite helpful in understanding what to try next and I'll try it today: https://www.siggraph.org/education/materials/HyperGraph/video/mpeg/mpegfaq/huffman_tutorial.html posted less than a minute ago by KarenWest	•••
Add a comment	

Showing all responses





are registered trademarks or to demarks of edX Inc.

© edX Inc. All rights reserved e ept where noted. EdX, Open edX and the edX and Open EdX logos

















