

## Tutorial 11: July 25th

1. For the following LZW problems, consider the initial dictionary to be the ASCII table.
  - a) Encode the following string using LZW: BANANA\_BANDANA
  - b) Decode the following encoded string using LZW:

71 – 73 – 86 – 69 – 95 – 77 – 131 – 82 – 69 – 128 – 137 – 65 – 83

2. Consider a modification to LZW to expand the dictionary faster: at every step, the encoder adds two new dictionary entries instead of one, when possible; one entry corresponds to the current string being encoded + the next character (like the usual LZW), while the other entry corresponds to the current string being encoded + the next two characters.

For example, if the text is APPLE, then after encoding A (as 65), the encoder adds two entries to the dictionary: AP at 128, and APP at 129. Note that after encoding L (as 76), the encoder only adds one new entry, LE at 134, and that no entries are added after encoding E (as 69).

- a) Encode the following string with the modified LZW: BAN\_ANANAS\_AND\_BANANAS.
- b) Modify the LZW decoding algorithm to decode strings that were coded with this modified LZW.
- c) Decode the following string that was encoded with the modified LZW:

82 – 79 – 84 – 79 – 95 – 77 – 69 – 138 – 78 – 133 – 147 – 128 – 143

3. Consider Burrows-Wheeler Transforms:
  - a) Encode the following string using BWT: MISSISSIPPI
  - b) Decode the following string using the inverse BWT: AIMOEOPN\$TOA