1. Short Answers

    a. Zipf's law shows that when a word is used more, the lower it would rank in a frequency table. Basically, the words that are used most in language do not describe that sentence at all, while words that are barely used can classify/describe a sentence much better. An example would be the word "the". It is the most used word in language, although it tells nothing about the parent sentence. A word such as "hypoxemia" would be used less, although it describes what the sentence is talking about much more. When using automatic text indexing, both types of words are removed from the word ranking, so that only significant words are indexed.

    b. By removing stop words from the index of a search engine, more focus can be placed on the significant words in the document. Frequently used words such as "the, as, a" can confuse the index system, as many documents contain these words – and in a high frequency. Removing stop words from the index can provide a more accurate search engine model.

    c. Removing stop words can impact simple searches, where there are not many significant words. For example, the search "to be or not to be". In this sentence there are not many significant words because many of the words would fall under the category of stop words – based on the frequency of using these words.

    d. Stemming is basically taking derived words and putting them into normal form. Say you have the word, "Create". Derived words could contain, "creative, creation, etc.". All these words are plural, adverbs, or inflected words, thus are derived words. To stem the word, the words would remove this, resulting in the word "creat".

    e. The index is called "inverted" because the documents are indexed twice. The first occurrence of indexing is ordering the terms with the document ID (and then sorting by term). Then the second occurrence of indexing, which makes it "inverted", is by splitting into dictionary and postings, with document frequency being added.

    f. Positional indexing is useful for solving phrase queries. It aids in removing the possibility of false positives in phrase queries. Positional indexing can be implemented for proximity queries, although it expands the storage needed.

2. Problem Solving

Doc 1: today you are you. That is truer than true.
Doc 2: you have brains in your head. You have feet in your shoes. You can steer yourself any direction you choose. You are on your own. And you know what you know. And you are the one who'll decide where to go.

a. **you:** 1: <2, 4>; 2: <1, 7, 13, 19, 21, 27, 30, 33>
head: 2: <6>
feet: 2: <9>

b. **"you are you":** 1: <2>
"head feet": none
"you /2 you": 1: <2>; 2: <19>

c. **New postings list structure:** docID: <sentenceID.pos1, sentenceID.pos2>
you: 1: <1.2, 1.4>