Web Search And Text Mining - A1
Anurag Kyal, 903040738

Solution 1.
a. fools rush in: 2,4,7
b. where angels fear: 4,7
c. fools rush in AND angels fear to tread: 4

Solution 2.
a. Assuming that $df_t \neq 0$ and assuming that we are dealing with a finite set of documents as it doesn't makes sense to consider a document set that is infinite in size, we have an upper and lower bound for idf_t. The upper bound is when the term appears only in one document when $idf_t = log(N)$ and the lower bound is when the term appears in each and every document when $idf_t = 0$. Since the document collection is finite, N is finite and since $0 \leq idf_t \leq log(N)$, the bound is finite.

b. df_t of such a term will be N and thus $idf_t = log(1) = 0$.
Using a stop word list allows us to eliminate common words like a, the, and etc from influencing the query results. The idf_t value also works on the same lines by making influence of a very commonly occurring word less compared to a less frequently occurring one and would do the work of stop words but it also has an added advantage of accounting the influence of words customized for the document collection. A particular document collection can have some words that are very common across documents and which should not have a big influence on the query but the stop word list is not customized to suit the needs of a document collection and will not allow this to happen.
However, we can't ignore the stop word list filtering even if we are using the idf_t in our calculation because not only will it require unnecessary place to store the stop words and spend additional processing power and time for irrelevant words which are to be ignored eventually, they might also result in irrelevant documents in case they end up having some influence in the query results.

c. idf values:

Term	idf value
car	log(806791/18165) = 1.648
moto	log(806791/6723) = 2.079
insurance	log(806791/19241) = 1.623
rent	log(806791/25235) = 1.505

tf-idf values:

TERM	DOC1	DOC2	DOC3
car	27*1.648 = 44.496	4*1.648 = 6.592	24*1.648 = 39.552
moto	3*2.079 = 6.237	33*2.079 = 68.607	0
insurance	0	33*1.623 = 53.559	29*1.623 = 47.067
rent	14*1.505 = 21.07	0	17*1.505 = 25.585

d. Yes, it can. Consider the case when in a collection of 100 documents, a word appears only in a

single document and the tf_d value for this document is 10. $tf - idf = 10 * log(100) = 100$.

Solution 3.
S1.
Q1:
Precision = 2/5 = 0.4
Recall = 2/4 = 0.5
F-Measure = 0.4444
Avg Precision(AP_1) = (1 + 2/3)/4 = 0.4166

Q2:
Precision = 2/5 = 0.4
Recall = 2/3 = 0.667
F-Measure = 0.5
Avg. Precision(AP_2) = (1 + 2/4)/3 = 0.5

MAP = $(AP_1 + AP_2)/2$ = 0.458

S2.
Q1:
Precision = 3/5 = 0.6
Recall = 3/4 = 0.75
F-Measure = 0.6667
Avg Precision(AP_1) = (1 + 1 + 3/5)/4 = 0.65

Q2:
Precision = 3/5 = 0.6
Recall = 3/3 = 1
F-Measure = .75
Avg. Precision(AP_2) = (1 + 2/3 + 3/4)/3 = 0.8055

MAP = $(AP_1 + AP_2)/2$ = 0.728

Precision, Recall and F Measure don't take rank into consideration while Average Precision and MAP do.

Solution 4.
a. Arithmetic mean of 2 numbers tends to take a value which is equidistant from the 2 numbers. So, in terms of information retrieval, In the 1st scenario, if the precision is too low (ex. 0.1) and the recall is very high (ex. 1.0), the mean score would be 0.55. In the second scenario, if the precision and recall are both okayish (ex. 0.55), the mean score would be still 0.55. If the system considers mean score, both the scenarios would be evaluated to perform equally good but the system in scenario 1 is not a desirable one.
Now, in the same setting as above, the harmonic mean for scenario 1 is 0.1818 and the harmonic mean for scenario 2 is 0.55. As is evident, the harmonic mean tends to be closer to the min of the 2 numbers and is a much better scoring scheme as if either precision or recall is too bad, it needs

to be penalized and the system should be scored low.

b. It can take any value in the range $0 - 1$.

c. There would be a break even point between precision and recall when the curve of precision vs. recall meets the curve $y = x$.
If a situation is desired where there would be no break even point between precision and recall, the precision-recall curve should be of the form $precsion = recall + c, c \neq 0$ which makes the precision-recall curve parallel to the curve $y = x$ and not equal to it or there should be a positive linear relationship between precision and recall.
However, precision = TP/TP+FP and recall = TP/TP+FN. As the number of documents retrieved increases, TP and FP increases while FN decreases. Thus, P decreases but recall increases. So, precision and recall clearly don't obey a linear relation and the curve tends to take the form of an inverse curve which is definitely bound to meet the curve $y = x$ at atleast one point in the domain 0 to ∞.