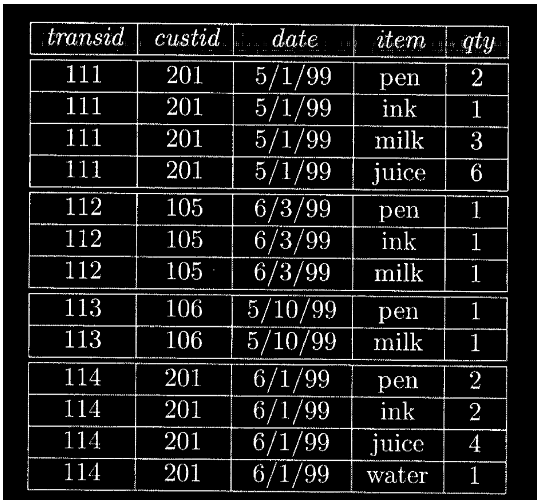
Homework 4 - Group 12

Name	
Name/email	Kelsey Neis (neis), Ashwin Sridhar (sridh052), Bela Demtchouk (derga002)

26.2 Consider the Purchases table shown in Figure 26.1.



transid	*custid*	*date*	*item*	*qty*
111	201	5/1/99	pen	2
111	201	5/1/99	ink	1
111	201	5/1/99	milk	3
111	201	5/1/99	juice	6
112	105	6/3/99	pen	1
112	105	6/3/99	ink	1
112	105	6/3/99	milk	1
113	106	5/10/99	pen	1
113	106	5/10/99	milk	1
114	201	6/1/99	pen	2
114	201	6/1/99	ink	2
114	201	6/1/99	juice	4
114	201	6/1/99	water	1

Figure 26.1 The Purchases Relation

1. Simulate the algorithm for finding frequent itemsets on the table in Figure 26.1 with minsup=90 percent, and then find association rules with minconf=90 percent.

Frequent Itemsets

1. {pen} is the only single item that appears in more than 90% of transactions. So, it is the only frequent item set, because no other items make it to the next iteration, per a Priori.

Association Rules

1. No association rules exist with the *minsup* and *minconf* values given

3. Simulate the algorithm for finding frequent itemsets of the table in Figure 26.1 with minsup=10 percent and then find association rules with minconf=90 percent.

Frequent Itemsets

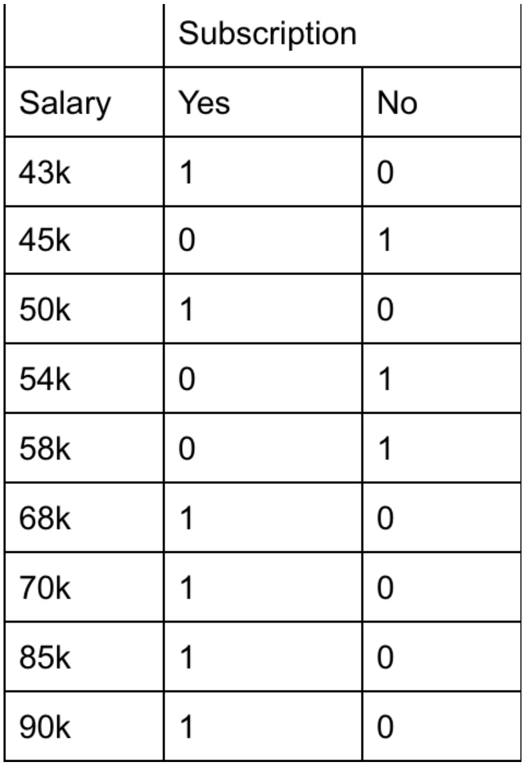
1. {pen}, {ink}, {milk}, {juice}, {water}
2. {pen, ink}, {pen, milk}, {pen, juice}, {pen, water}, {ink, milk}, {ink, juice}, {ink, water}, {milk, juice}, {juice, water}
3. {pen, ink, milk}, {pen, ink, juice}, {pen, ink, water}, {pen, juice, water}, {ink, milk, juice}, {ink, juice, water}
4. {pen, ink, milk, juice}, {pen, ink, juice, water}

Association Rules

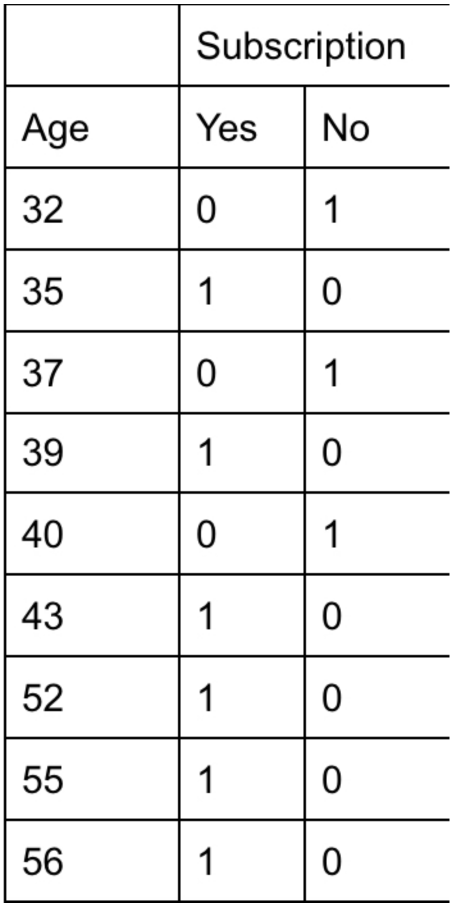
1. {ink, milk, juice} → {pen}
2. {ink, juice, water} → {pen}
3. {milk} → {pen}
4. {ink} → {pen}
5. {juice} → {pen}
6. {water} → {pen}

Question 26.8 - Consider the SubscriberInfo Relation shown in Figure 26.17. It contains information about the marketing campaign of the DB Aficionado magazine. The first two columns show the age and salary of a potential customer and the subscription column shows whether the person subscribes to the magazine. We want to use this data to construct a decision tree that helps predict whether a person will subscribe to the magazine.

1. Construct the AVC-group of the root node of the tree.

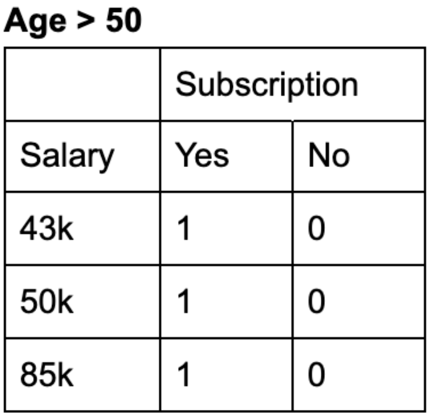


	Subscription	
Salary	Yes	No
43k	1	0
45k	0	1
50k	1	0
54k	0	1
58k	0	1
68k	1	0
70k	1	0
85k	1	0
90k	1	0



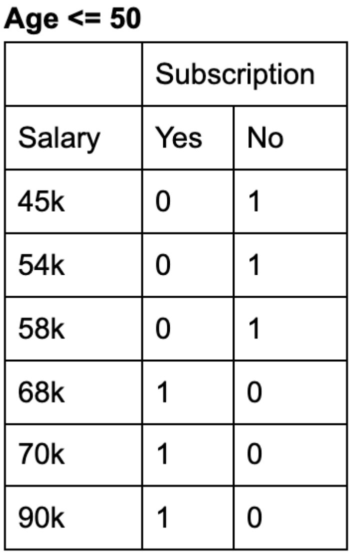
	Subscription	
Age	Yes	No
32	0	1
35	1	0
37	0	1
39	1	0
40	0	1
43	1	0
52	1	0
55	1	0
56	1	0

2. Assume that the splitting predicate at the root node is age ≤ 50. Construct the AVC-groups of the two children nodes of the root node.



Age > 50

	Subscription	
Salary	Yes	No
43k	1	0
50k	1	0
85k	1	0



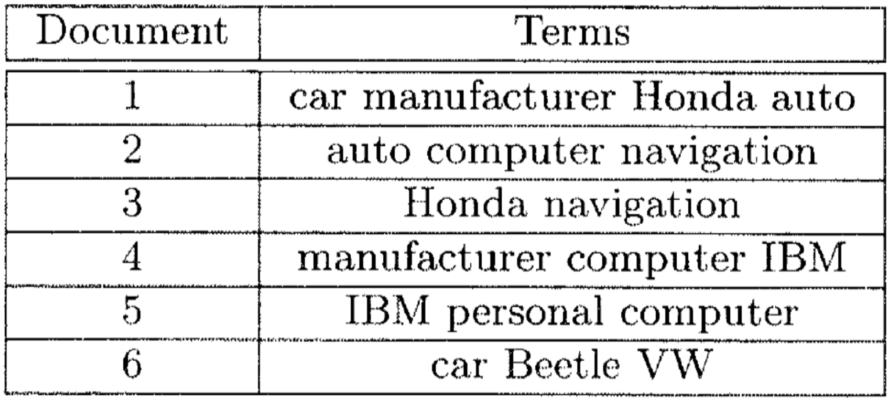
Age <= 50

	Subscription	
Salary	Yes	No
45k	0	1
54k	0	1
58k	0	1
68k	1	0
70k	1	0
90k	1	0

Question 26.10 - Assume you are given the three sequences <1, 3, 4>, <2, 3, 2>, <3, 3, 7>. Compute the Euclidian Norm between all pairs of sequences.

- **<1, 3, 4> and <2, 3, 2>**: $(1-3)^2+(3-3)^2+(4-2)^2=8$
- **<1, 3, 4> and <3, 3, 7>:** $(1-3)^2+(3-3)^2+(4-7)^2=13$
- **<2, 3, 2> and <3, 3, 7>**: $(2-3)^2+(3-3)^2+(2-7)^2=26$

Question 27.2 - Assume you are given a document database that contains SIX documents. After stemming, the documents contain the following terms:



Document	Terms
1	car manufacturer Honda auto
2	auto computer navigation
3	Honda navigation
4	manufacturer computer IBM
5	IBM personal computer
6	car Beetle VW

1. Show the result of creating an inverted file on the documents.

Inverted file

Term	docid[tf], docid[tf] ...
car	1[1], 6[1]
manufacturer	1[1], 4[1]
Honda	1[1], 3[1]
auto	1[1], 2[1]
computer	2[1], 4[1], 5[1]
navigation	2[1], 3[1]
IBM	4[1], 5[1]
personal	5[1]
Beetle	6[1]

Term	docid[tf], docid[tf] ...
VW	6[1]

3. Evaluate the following boolean queries using the inverted file and the signature file that you created: 'car', 'IBM' AND 'Computer', 'IBM' AND 'car', 'IBM' OR 'auto', and 'IBM' AND 'computer' AND 'manufacturer'.

- **car:** 1, 6
- **IBM AND Computer:** 4, 5
- **IBM AND car:** $\emptyset$
- **IBM OR auto:** 1, 2, 4, 5
- **IBM AND Computer AND manufacturer:** 4

5. Consider the following ranked queries: 'car', 'IBM Computer', 'IBM car', 'IBM auto', and 'IBM Computer manufacturer'.

$$IDF = \log_2 \frac{N}{n_j}$$

Where N is the total number of documents and n_j = the number of documents that the term j appears in.

(a) Calculate the IDF for every term in the database. (Assuming log base 2)

- car: $log(6/2) = log(3) = 1.58$
- manufacturer: $log(6/2) = log(3) = 1.58$
- Honda: $log(6/2) = log(3) = 1.58$
- auto: $log(6/2) = log(3) = 1.58$
- computer: $log(6/3) = log(2) = 1$
- navigation: $log(6/2) = log(3) = 1.58$
- IBM: $log(6/2) = log(3) = 1.58$
- personal: $log(6/1) = log(6) = 2.58$
- Beetle: $log(6/1) = log(6) = 2.58$
- VW: $log(6/1) = log(6) = 2.58$

(b) For each document, show its document vector.

$w_{ij} = t_{ij} * IDF$ is the weighted formula. Table below is NOT weighted.

Document vectors

Docid	car	manufacturer	Honda	auto	computer	navigation	IBM	personal	Beetle	VW
1	1	1	1	1						
2				1	1	1				
3			1			1				
4		1			1		1			
5					1		1	1		
6	1								1	1

(c) For each query, calculate the relevance of each document in the database, with and without the length normalization step.

document similarity: $sim(Q, D_i) = \sum_{j=1}^{t} w_{qj} * w_{d_{ij}} * IDF$

with normalization:

$$sim(Q, D_i) = \frac{\sum_{j=1}^{t} w_{qj} * w_{d_{ij}} * IDF_j}{\sqrt{\sum_{j=1}^{t} w_{qj}^2 * \sum_{j=1}^{t} w_{d_{ij}}^2}}$$

- **car:**
  - TF: 1 → Q(1)
  - Document 1:
    - TF: 1 → $D(1)$
    - *without normalization:* $1 * 1 * 1.58 = 1.58$
    - *with normalization:* $1.58/\sqrt{1^2 * 1^2} = 1.58$
  - Document 2: 0, Document 3: 0, Document 4: 0, Document 5: 0
  - Document 6:
    - TF: 1 → $D(1)$
    - *without normalization*: $1 * 1 * 1.58 = 1.58$

        - *with normalization*: $1.58/\sqrt{1^2 * 1^2} = 1.58$
- **IBM computer:**
    - TF: IBM: 1; computer: 1 → $Q(1,1)$
    - Document 1: 0
    - Document 2:
        - TF: IBM: 0; computer: 1 → $D(0,1)$
        - *without normalization*: $1 * 0 * 1.58 + 1 * 1 * 1 = 1$
        - *with normalization*: $1/\sqrt{[1^2 + 1^2] * [0 + 1^2]} = 0.5$
    - Document 3: 0
    - Document 4:
        - TF: IBM: 1; computer: 1 → $D(1,1)$
        - *without normalization*: $(1 * 1) * 1.58 + (1 * 1) * 1 = 2.58$
        - *with normalization*: $2.58/\sqrt{[1^2 + 1^2] * [1^2 + 1^2]} = 1.29$
    - Document 5:
        - TF: IBM: 1; computer: 1 → $D(1,1)$
        - *without normalization*: $(1 * 1) * 1.58 + (1 * 1) * 1 = 2.58$
        - *with normalization*: $2.58/\sqrt{[1^2 + 1^2] * [1^2 + 1^2]} = 1.29$
    - Document 6: 0
- **IBM car:**
    - TF: IBM: 1; car: 1 → $Q(1,1)$
    - Document 1:
        - TF: IBM: 0; car: 1 → $D(0,1)$
        - *without normalization*: $1 * 0 + (1 * 1) * 1.58 = 1.58$
        - *with normalization*: $1.58/\sqrt{[1^2 + 1^2] * [0 + 1^2]} = 1.11$
    - Document 2: 0, Document 3: 0
    - Document 4:
        - TF: IBM: 1; car: 0 → $D(1,0)$

        - *without normalization*: $(1*1)*1.58+(1*0)=1.58$
        - *with normalization*: $1.58/\sqrt{[1^2+1^2]*[0+1^2]}=1.11$
    - Document 5:
        - TF: IBM: 1; car: 0 → $D(1,0)$
        - *without normalization*: $(1*1)*1.58+(1*0)*1.58=1.58$
        - *with normalization*: $1.58/\sqrt{[1^2+1^2]*[0+1^2]}=1.11$
    - Document 6:
        - TF-IDF: IBM: 0; car: 1 ⇒ D(0, 1)
        - *without normalization*: $(1*0)*1.58+(1*1)*1.58=1.58$
        - *with normalization*: $1.58/\sqrt{[1^2+1^2]*[0^2+1^2]}=1.11723$
- **IBM auto:**
    - TF-IDF: IBM: 1; auto: 1 → $Q(1,1)$
    - Document 1:
        - TF: IBM: 0; auto: 1 → $D(0,1)$
        - *without normalization*: $(1*0)*1.58+(1*1)*1.58=1.58$
        - *with normalization*: $1.58/\sqrt{[1^2+0^2]*[1^2+1^2]}=1.11723$
    - Document 2:
        - TF: IBM: 0; auto: 1 → $D(0,1)$
        - *without normalization*: $(1*0)*1.58+(1*1)*1.58=1.58$
        - *with normalization*: $1.58/\sqrt{[1^2+0^2]*[1^2+1^2]}=1.11723$
    - Document 3: 0
    - Document 4:
        - TF: IBM: 1; auto: 0 → $D(1,0)$
        - *without normalization*: $(1*1)*1.58+(1*0)*1.58=1.58$
        - *with normalization*: $1.58/\sqrt{[1^2+1^2]*[0+1^2]}=0.11723$
    - Document 5:
        - TF: IBM: 1; auto: 0 → $D(1,0)$

        - *without normalization*: $(1*1)*1.58+(1*0)*1.58=1.58$
        - *with normalization*: $1.58/\sqrt{[1^2+1^2]*[0+1^2]}=0.11723$
    - Document 6: 0
- **IBM computer manufacturer**
    - TF: IBM: 1; computer: 1; manufacturer: 1 → $Q(1,1,1)$
    - Document 1:
        - TF: IBM: 0; computer: 0; manufacturer: 1 → $D(0,0,1)$
        - *without normalization*: $sim(Q,D)=(1*0)+(1*0)+(1*1)*1.58=1.58$
        - *with normalization*: $\frac{1.58}{\sqrt{[1^2+1^2+1^2]*[0^2+0^2+1^2]}}=0.91$
    - Document 2:
        - TF: IBM: 0; computer: 1; manufacturer: 0 → $D(0,1,0)$
        - *without normalization*: $(1*1)*1=1$
        - *with normalization*: $\frac{1}{\sqrt{[1^2+1^2+1^2]*[0^2+1^2+0^2]}}=0.577$
    - Document 3: 0
    - Document 4:
        - TF: IBM: 1; computer: 1; manufacturer: 1 → $D(1,1,1)$
        - *without normalization*: $(1*1)*1.58+(1*1)*1+(1*1)*1.58=4.16$
        - with normalization: $\frac{4.16}{\sqrt{[1^2+1^2+1^2]*[1^2+1^2+1^2]}}=1.38$
    - Document 5:
        - TF: IBM: 1; computer: 1; manufacturer: 0 → $D(1,1,0)$
        - *without normalization*: $(1*1)*1.58+(1*1)*1+(1*0)=2.58$
        - with normalization: $\frac{2.58}{\sqrt{[1^2+1^2+1^2]*[1^2+1^2+0]}}=1.053$
    - Document 6: 0

**(d) Describe how you would use the inverted index to identify the top two documents
that match each query.**

I would use the inverted index to retrieve the documents that contain the query terms, then compute their TF*IDF values to rank the documents by relevance.

I would sort the postings file of the inverted index and return first 2 entries of each inverted list.

(e) How would having the inverted lists sorted by relevance instead of document id
affect your answer to the previous question?

Assuming the relevance is sorted descending, the first two results of inverted list would represent the most relevant documents to the query. If inverted list is sorted I would just return the first two results.

(f) Replace each document with a variation that contains 10 copies of the same document. For each query, recompute the relevance of each document, with and without the length normalization step.

- **car:**
    - TF: $1 = Q(1)$
    - Document 1:
        - TF: $10 = D(10)$
        - *without normalization:* $(10 * 1) * 1.58 = 15.8$
        - *with normalization:* $15.8/\sqrt{10^2 * 1^2} = 1.58$
    - Document 2: 0, Document 3: 0, Document 4: 0, Document 5: 0
    - Document 6:
        - TF: $10 = D(10)$
        - *without normalization:* $(10 * 1) * 1.58 = 15.8$
        - *with normalization*: $15.8/\sqrt{10^2 * 1^2} = 1.58$
- **IBM computer:**
    - TF: IBM: 1; computer: 1 → $Q(1, 1)$
    - Document 1: 0
    - Document 2:
        - TF: IBM: 0; computer: $10 * 1 = 1$ → $D(0, 10)$
        - *without normalization*: $(0 * 1) + (10 * 1) * 1 = 10$

        - *with normalization*: $10/\sqrt{[1^2+1^2]*[0+10^2]} = 0.707$
    - Document 3: 0
    - Document 4:
        - TF: IBM: 10; computer: 10 → $D(10,10)$
        - *without normalization*: $(10*1)*1.58+(1*10)*1 = 25.8$
        - *with normalization*: $25.8/\sqrt{[1^2+1^2]*[10^2+10^2]} = 1.29$
    - Document 5:
        - TF: IBM: 10; computer: 10 → $D(10,10)$
        - *without normalization*: $(1*10)1.58+(1*10)*1 = 25.8$
        - *with normalization*: $25.8/\sqrt{[1^2+1^2]*[10^2+10^2]} = 1.29$
    - Document 6: 0
- **IBM car: BELA**
    - TF: IBM: 1; car: 1 → $Q(1,1)$
    - Document 1:
        - TF: IBM: 0; car: 10 → $D(0,10)$
        - *without normalization*: $1.58*(1*0)+(1.58*(1*10)) = 15.8$
        - *with normalization*: $15.8/\sqrt{[1*1]*[0+10^2]} = 1.58$
    - Document 2: 0, Document 3: 0
    - Document 4:
        - TF: IBM: 0; car: 10 → $D(0,10)$
        - *without normalization*: $1.58*(1*0)+(1.58*(1*10)) = 15.8$
        - *with normalization*: $15.8/\sqrt{[1*1]*[0+10^2]} = 1.58$
    - Document 5:
        - TF: IBM: 10; car: 0 → $D(10,0)$
        - *without normalization*: $1.58*(1*0)+(1.58*(1*10)) = 15.8$
        - *with normalization*: $15.8/\sqrt{[1*1]*[0+10^2]} = 1.58$
    - Document 6:

    - TF: IBM: 0; car: 10 → $D(0,10)$
    - *without normalization*: $1.58 * (1 * 0) + (1.58 * (1 * 10)) = 15.8$
    - *with normalization*: $15.8/\sqrt{[1 * 1] * [0 + 10^2]} = 1.58$
- **IBM auto:**
  - TF: IBM: 1; auto: 1 → $Q(1,1)$
  - Document 1:
    - TF: IBM: 0; auto: 10 → $D(0,10)$
    - *without normalization*: $(1 * 0) + (1 * 10) * 1.58 = 15.8$
    - *with normalization*: $15.8/\sqrt{[1^2 + 1^2] * [0 + 10^2]} = 1.11723$
  - Document 2:
    - TF: IBM: 0; auto: 10 → $D(0,10)$
    - *without normalization*: $(1 * 0) + (10 * 1) * 1.58 = 15.8$
    - *with normalization*: $15.8/\sqrt{[1^2 + 1^2] * [0 + 10^2]} = 1.11723$
  - Document 3: 0
  - Document 4:
    - TF: IBM: 10; auto: 0 → $D(10,0)$
    - *without normalization*: $(1 * 0) + (10 * 1) * 1.58 = 15.8$
    - *with normalization*: $15.8/\sqrt{[1^2 + 1^2] * [0 + 10^2]} = 1.11723$
  - Document 5:
    - TF: IBM: 10; auto: 0 → $D(10,0)$
    - *without normalization*: $(1 * 0) + (10 * 1) * 1.58 = 15.8$
    - *with normalization*: $15.8/\sqrt{[1^2 + 1^2] * [0 + 10^2]} = 1.11723$
  - Document 6: 0
- **IBM computer manufacturer — ASHWIN**
  - TF: IBM: 1; computer: 1; manufacturer: 1 → $Q(1,1,1)$
  - Document 1:
    - TF: IBM: 0; computer: 0; manufacturer: 10 → $D(0,0,10)$

    - *without normalization*: $sim(Q,D) = (1*0) + (1*0) + (10*1*1.58) = 15.8$
    - *with normalization*: $\frac{15.8}{\sqrt{[1^2+1^2+1^2]*[0^2+0^2+10^2]}} = 0.912213$
- Document 2:
    - TF: IBM: 0; computer: 10; manufacturer: 0 → $D(0,10,0)$
    - *without normalization*: $10*1*1 = 10$
    - *with normalization*: $\frac{10}{\sqrt{[1^2+1^2+1^2]*[0^2+10^2+0^2]}} = .577$
- Document 3: 0
- Document 4:
    - TF: IBM: 10; computer: 10; manufacturer: 10 → $D(10,10,10)$
    - *without normalization*: $(10*1)*1.58 + (10*1)*1 + (10*1)*1.58 = 41.6$
    - with normalization: $\frac{41.6}{\sqrt{[1^2+1^2+1^2]*[10^2+10^2+10^2]}} = 1.386$
- Document 5:
    - TF: IBM: 10; computer: 10; manufacturer: 0 → $D(10,10,0)$
    - *without normalization*: $(10*1*1.58) + (10*1*1.58) + (1*0) = 31.6$
    - with normalization: $\frac{31.6}{\sqrt{[1^2+1^2+1^2]*[10^2+10^2+0]}} = 1.29$
- Document 6: 0

27.4 You are in charge of the Genghis ('We execute fast') search engine. You are designing your server cluster to handle 500 million hits a day and 10 billion pages of indexed data. Each machine costs $1000, and can store 10 million pages and respond to 200 queries per second (against these pages).

1. If you were given a budget of $500,000 dollars for purchasing machines, and were required to index all 10 billion pages, could you do it?

$cost = \frac{10000000000 totalPages}{10000000 capacityPerMachine} * 1000 costPerMachine$

No, $500,000 would only be enough to buy 500 machines at $1000 each. Since each machine can store 10 million pages, 500 machines would only be enough to store 5 billion pages, which is half the amount needed. I would need $1,000,000, enough to purchase 1000 machines.

2. What is the minimum budget to index all pages? If you assume that each query can be answered by looking at data in just one (10 million page) partition, and that queries are uniformly distributed across partitions, what peak load (in number of queries per second) can such a cluster handle?

- Minimum budget: $1,000,000 (see part 1 above)
- The cluster can handle a peak load of 200,000 queries per second. Minimum number of machines is 1000, each can handle 200 queries. Thus top load is 1000*200

3. How would your answer to the previous question change if each query, on average, accessed two partitions?

The peak load would be halved, 100,000

4. What is the minimum budget required to handle the desired load of 500 million hits per day if all queries are on a single partition? Assume that queries are uniformly distributed with respect to time of day.

number of seconds in a day = 86400s

Number of queries per day for one machine = 86400s/d*200q/s=17,280,000q/d

Number of machines required= 500,000,000/17,280,000=29

Cost=29*1000= $29,000

5. How would your answer to the previous question change if the number of queries per day went up to 5 billion hits per day? How would it change if the number of pages went up to 100 billion?

- $290,000
- Assuming the new number of 100 billion is just the number of initial indexing, the increase of pages wouldn't affect the budget for serving the queries.

Since the number of queries incoming into each machine did not change and the capacity of queries that each machine can handle did not change there would be no change in the budget either.

- If the question is asking how would the increase in the number of pages served (hits) affect the ongoing budget, then it would increase to (100,000,000,000/17,280,000)*1000= $5,787,000

6. Assume that each query accesses just one partition, that queries are uniformly distributed across partitions, but that at any given time, the peak load on a partition is up to 10 times the average load. What is the minimum budget for purchasing machines in this scenario?

Assuming the queries span the whole indexed internet of 10 billion pages.

new peak load that needs to be supported = 500 mil * 10 = 5 billion

number of seconds in a day = 86400s

Number of queries per day for one machine = 86400s/d*200q/s=17,280,000q/d

Number of machines required= 5 000,000,000/17,280,000=290

Cost=290*1000= $290,000

7. Take the cost for machines from the previous question and multiply it by 10 to reflect the costs of maintenance, administration, network bandwidth, etc. This amount is your annual cost of operation. Assume that you charge advertisers 2 cents per page. What fraction of your inventory (i.e., the total number of pages that you serve over the course of a year) do you have to sell in order to make a profit?

annual op cost = $290 000 * 10 = $2 900 000

$2 900 000 /0.02 $/page = 145000000 pages

145000000 / 10 bill = 0.0145 ⇒ > 1.45% of pages need to be sold to break even