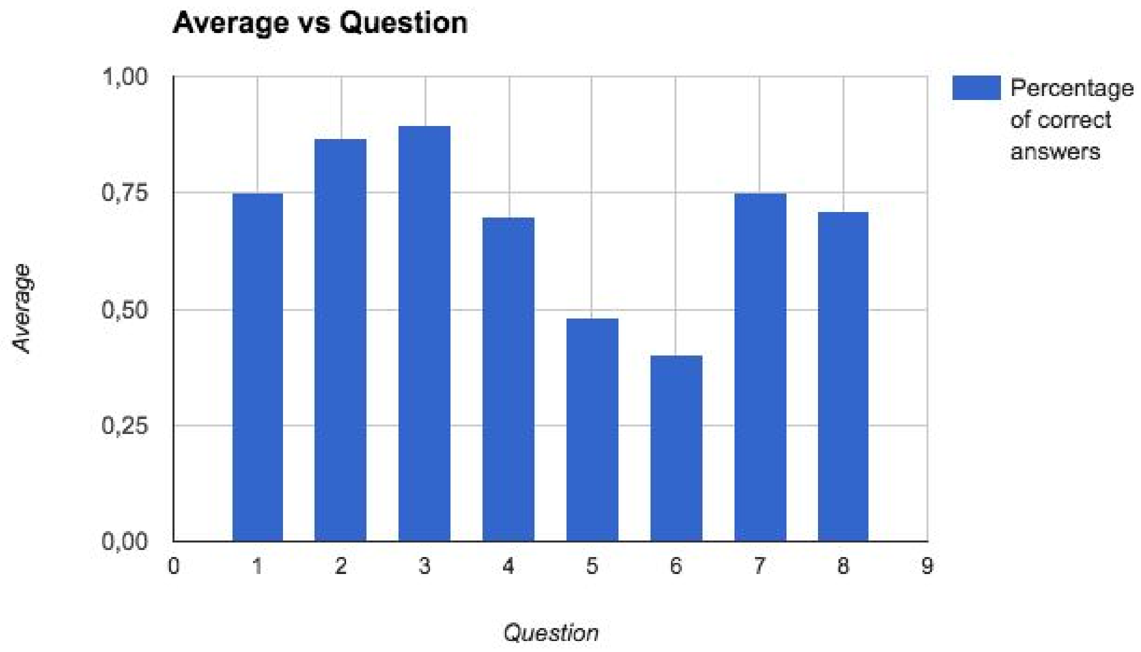
# Quiz 2

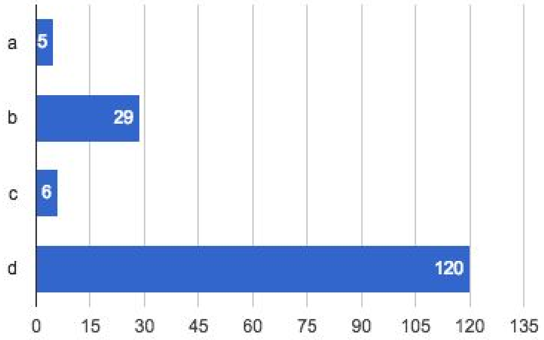
Solutions


Average vs Question
1,00
0,75
0,50
0,25
0,00
Average
0
1
2
3
4
5
6
7
8
9
Question
Percentage of correct answers

# Question 1

Given the 2-itemsets {1, 2}, {1, 3}, {2, 3}, {2, 5}, {3, 5}, when generating the 3-itemset we will:

- ☐ a. Have **4** 3-itemsets after the **join** and **4** 3-itemsets after the **prune**
- ☐ b. Have **4** 3-itemsets after the **join** and **2** 3-itemsets after the **prune**
- ☐ c. Have **3** 3-itemsets after the **join** and **3** 3-itemsets after the **prune**
- ☒ d. Have **2** 3-itemsets after the **join** and **2** 3-itemsets after the **prune**




**join**

**prune**

{1, 2}
{1, 3}
{2, 3}
{2, 5}
{3, 5}

{1, 2, 3}
{2, 3, 5}

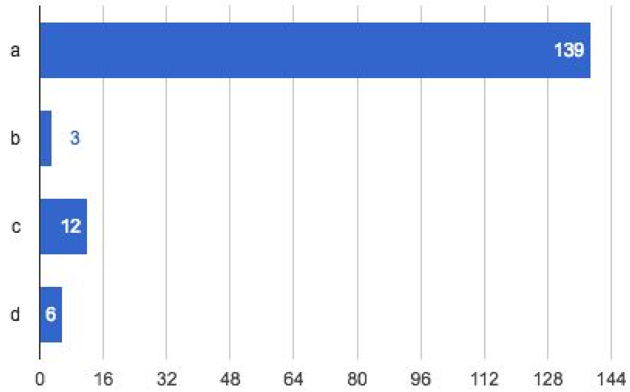
join sets with same prefix

{1, 2}, {1, 3}, {2, 3} in 2-itemsets

{2, 3}, {2, 5}, {3, 5} in 2-itemsets

{1, 2, 3}
{2, 3, 5}

# Question 2




Given the following transactions {milk, bread}, {eggs, bread}, {milk, eggs, bread}, {eggs}, {milk, eggs}, {milk}

- [x] a. bread⇒milk has support ⅓ and confidence ⅔
- [ ] b. eggs⇒milk has support ⅓ and confidence ⅔
- [ ] c. milk⇒bread has support ⅓ and confidence ⅔
- [ ] d. milk⇒eggs has support ⅓ and confidence ⅔

**support(bread⇒milk) = p({milk, bread}) = c({milk, bread})/N = 2/6 = ⅓**
**confidence(bread⇒milk) = p(milk | bread) = c({milk, bread})/c(bread) = ⅔**

support(eggs⇒milk) = p({milk, eggs}) = c({milk, eggs})/N = 2/6 = ⅓
confidence(eggs⇒milk) = p(milk | eggs) = c({milk, eggs})/c(eggs) = 2/4 = ½

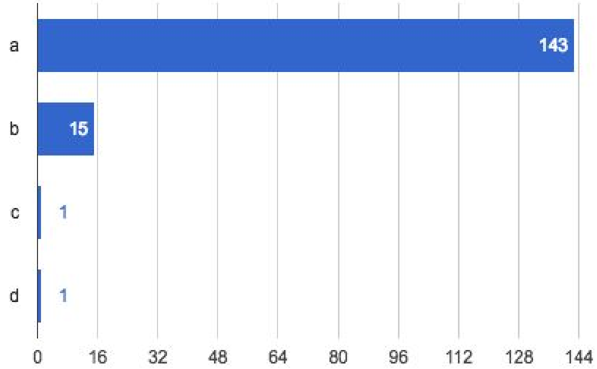
support(milk⇒bread) = p({bread, milk}) = c({bread, milk})/N = 2/6 = ⅓
confidence(milk⇒bread) = p(bread | milk) = c({bread, milk})/c(milk) = 2/4 = ½

support(milk⇒eggs) = p({milk, eggs}) = c({milk, eggs})/N = 2/6 = ⅓
confidence(milk⇒eggs) = p(eggs | milk) = c({milk, eggs})/c(milk) = 2/4 = ½

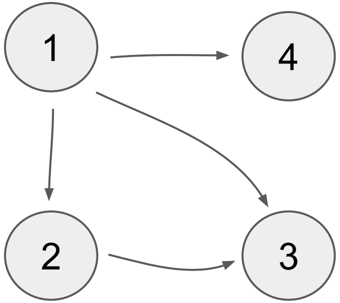
# Question 3

If we have a graph with nodes {1, 2, 3, 4} and edges {1→2, 1→3, 1→4, 2→3} then the authority values, without normalization, are:

- [x] a. (0, ¼, ½, ¼)
- [ ] b. (¾, ¼, 0, 0)
- [ ] c. (0, ⅓, ⅓, ⅓)
- [ ] d. (½, ½, 0, 0)




0 incoming edges
out of 4 edges




1 incoming edge
out of 4 edges

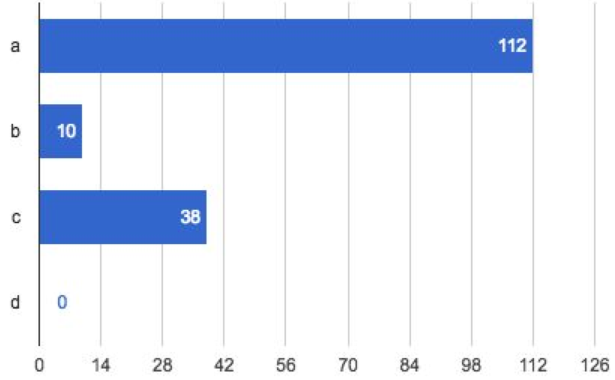
1 incoming edge
out of 4 edges

2 incoming edges
out of 4 edges

# Question 4

If milk⇒{bread, eggs} has confidence c1 and milk⇒bread has confidence c2, then:

- ☒ a. c1 <= c2
- ☐ b. c2 <= c1
- ☐ c. c1 < c2 and c2 < c1 are possible
- ☐ d. c1 = c2



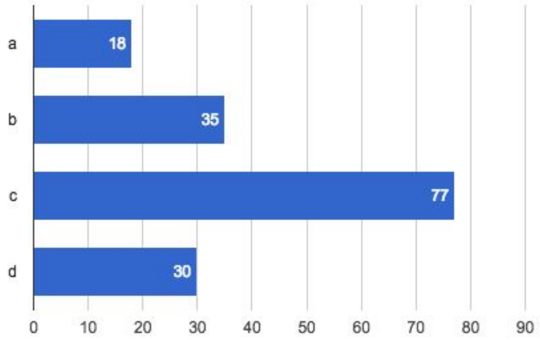

c1 = confidence(milk⇒{bread, eggs})
= p({bread, eggs, milk} | milk)
= p({bread, eggs, milk}) / p(milk)

c2 = confidence(milk⇒bread)
= p({bread, milk} | milk)
= p({bread, milk}) / p(milk)

then

c1 = p({bread, eggs, milk}) / p(milk) **<=** p({bread, milk}) / p(milk) = c2

# Question 5



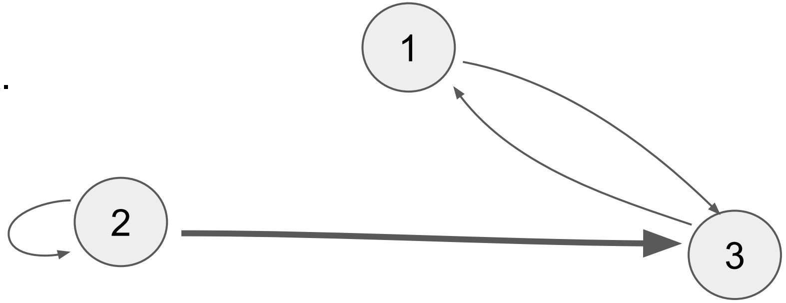

Given the following matrix for teleporting in a random walker model:

$$\begin{bmatrix} 0 & 0 & 1 \\ 0 & \frac{1}{2} & 0 \\ 1 & \frac{1}{2} & 0 \end{bmatrix}$$

Which of the following is true (**independent of how the link matrix is given**):

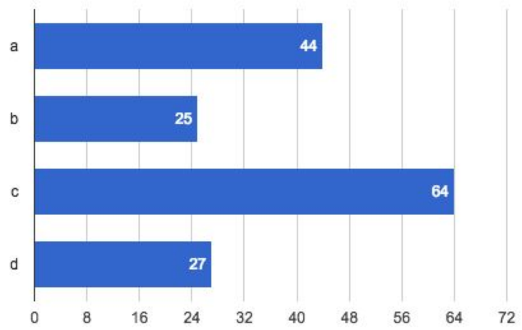
- [ ] a. A random walker can always reach node 2 (if no links, can't go from 1 to 2)
- [ ] b. A random walker can always reach any node (if no links, can't go from 1 to 2 or 3 to 2)
- [x] c. A random walker can always leave node 2 (can leave through the fat edge)
- [ ] d. A random walker can never reach node 2 (if all links exist any node can be reached, in particular node 2)

Note that the statement must hold for any link matrix.



# Question 6

Which of the following statements concerning compression of adjacency lists for link indexing **is wrong**:




- ☐ a. Compression can exploit the fact that most links of a page point to the page itself
- ☐ b. Compression can exploit the fact that pages with similar URLs typically have also many outgoing links in common
- ☒ c. Exploiting similarity among different adjacency lists will always decrease the cost of encoding of adjacency lists
- ☐ d. Compression works well, even if we consider similarity of adjacency lists only for a fraction of neighbouring URLs in the lexicographically order

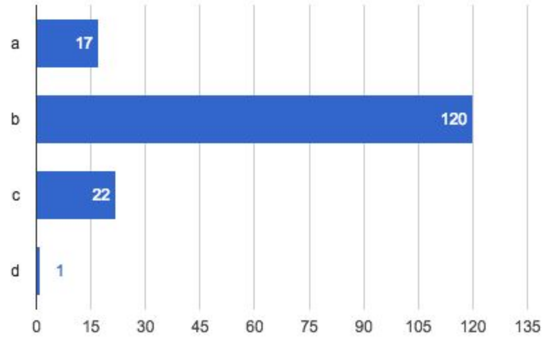
Why?

If there are no similarity between the adjacency lists and we add the additional data to be able to compress similar lists will increase the cost. Assumptions say that this is not the case in practice, but it is still possible

# Question 7

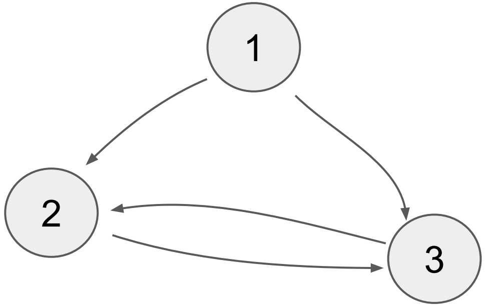
Given the graph 1→2, 1→ 3, 2→3, 3→2, the *Page Rank* value of this graph is (without random jumps)

- ☐ a. (0, 1, 1)
- ☒ b. (0, ½, ½)
- ☐ c. (⅕, ⅖, ⅖)
- ☐ d. (1, 0, 0)



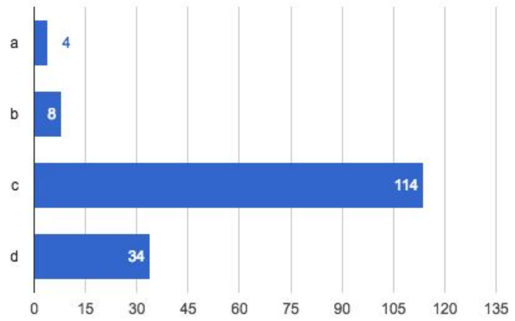

$$L = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}$$

$$R = \begin{bmatrix} 0 & 0 & 0 \\ 1/2 & 0 & 1 \\ 1/2 & 1 & 0 \end{bmatrix}, \quad p = \begin{bmatrix} 0 \\ 1/2 \\ 1/2 \end{bmatrix}$$




See slide 16-17 of Link Analysis

# Question 8




When computing PageRank iteratively the computation ends when

- ☐ a. The norm of the rank vector exceeds a predefined threshold
- ☐ b. All nodes of the graph have been visited a predefined number of times
- ☒ c. The norm of the difference of rank vectors of two subsequent iterations falls below a predefined threshold
- ☐ d. The difference among the Eigenvalues of two subsequent iterations falls below a predefined threshold

See algorithm in slide 22 of Link Analysis

The norm $\delta$ of the difference of rank vectors of two subsequent iterations ($\mathbf{p}_{i+1}$ and $\mathbf{p}_i$) falls below a predefined threshold $\varepsilon$.