${\bf Big~Data~Management~(MIRI/MDS/BDMA)}$

Fan	nily name:	Given name:	
ans	(20%) Compare a B-tree and an LSM-tree in the context of the RUM conjecture (i.e., as an answer to this question, three brief explanations of the form "From the perspective of X, Y-tree is better than Z-tree, because of this and that." are expected).		
(a)) R		
(b)) U		
(6)	,		
(c)) M		
con con kin	taining $16 \cdot 10^5$ rows in a Parqu stant" and this attribute contains d of compression has been disable	raw data stored in an HDFS cluster of 50 machines, and et file ; consider you have a query over an attribute " $A =$ s only 100 different and equiprobable values. Assuming any ed, explicit any assumption you need to make and give the nt metadata) it would need to fetch from disk.	
•	• Replication factor: 3 (default)		
•	• Chunk size: 128MB (default)		
•	• Rowgroup size: $32MB$		
• • •			

3.	to be	6) Given an empty Consistent Hash with $h(x) = x\%32$ (i.e., we directly take module 32 of the keys and the bucket IDs), and unlimited capacity in each bucket, consider you have after of four machines with IDs 19, 22, 75, 92, and draw the result of inserting the following in the given order: 12, 4, 10, 49, 42, 60, 63, 53, 47, 27, 26, 28, 13, 52.
4.	,	$\stackrel{(6)}{}$ Assume you have a MongoDB collection which occupies 6 chunks UNevenly distributed
		shards (i.e., 1, 2 and 3 chunks per shard respectively). Being the document Id also the shard the chunk of a document is determined by means of a hash function. Assuming that
		ssing one document takes one time unit (existing indexes are used at no cost) and we have
		0 documents in the collection, k of which have value "YYY" for attribute "other", how many units would take the following operations ¹ :
		FindOne($\{ \text{-id} : \text{"}XXX\text{"}\} $)
	(**)	
	(b)	$Find(\{ id : \{ in : [1,,3000] \} \})$, being $[1,,6000]$ the range of existing IDs.
	(c)	$Find(\{other: "YYY"\})$, being the attribute indexed.
	(0)	Tima((other: 111)), being the attribute indexed.
	(d)	$Find(\{other: "YYY"\})$, being the attribute NOT indexed.

¹As typically in RDBMS optimizers, assume uniform distribution of values and statistical independence between pairs of attributes.

${\bf Big~Data~Management~(MIRI/MDS/BDMA)}$

Final Exam

	Family name:	Given name:		
5	5. (20%) Given two files containing the following ki	nds of data:		
ο.	Employees.txt with fields: EmployeeID; EmployeeNam EMP4;RICARDO;250000;MADRID;DPT4 EMP5;EULALIA;150000;BARCELONA;DPT5 EMP6;MIQUEL;125000;BADALONA;DPT5 EMP7;MARIA;175000;MADRID;DPT6 EMP8;ESTEBAN;150000;MADRID;DPT6			
	Departments.txt with fields: SiteID; DepartmentName DPT1;DIRECCIO;10;PAU CLARIS;BARCELONA DPT2;DIRECCIO;8;RIOS ROSAS;MADRID DPT3;MARKETING;1;PAU CLARIS;BARCELON DPT4;MARKETING;3;RIOS ROSAS;MADRID			
	Consider the following PySpark code and answ	ver the questions bellow.		
	2: B = sc.textFile("departments.t 3: C = A.map(lambda t: t[3]) 4: D = B.map(lambda t: t[1]) 5: E = D.cartesian(C) 6: F = B.map(lambda t: (t[1], t[4]) 7: G = E.subtract(F) 8: H = G.map(lambda t: t[0]) 9: R = D.subtract(H)			
	(a) State in natural language the corresponding query it would answer?			
	(b) Clearly indicate any mistake or improvement give (1) the line number, (2) pseudo-code to	t you can fix/make in the code? For each of them implement the fix, and (3) brief rationale.		