BDA (Sheet-2) Group- C7

(Prince, Lyon, Bhautik)

Ans.-1 We have to perform LSH on given signature matrix.

here we're using LSH with three bands of two rows each.

	C_1	C_2	C_3	C_4	C_5	C_6	C_7
Band-1	1	2	1	1	2	5	4
	2	3	4	2	3	2	2
Band-2	3	1	2	3	1	3	2
	4	1	3	1	2	4	4
Band-3	5	2	5	1	1	5	1
	6	1	6	4	1	1	4

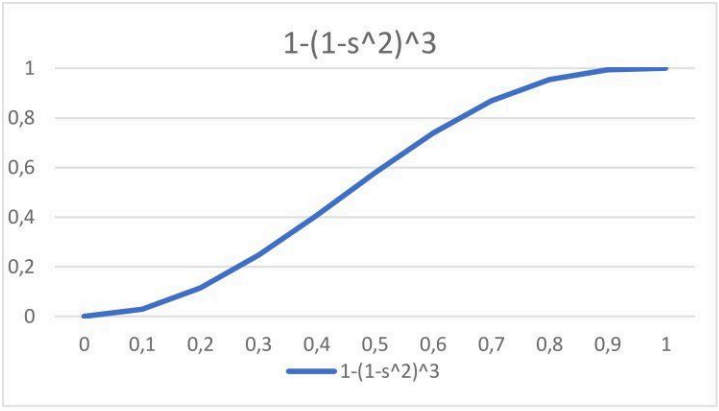
here; the Hash function is just the identity so we just look for identical pairs on each band, for all columns c_i and c_j, $i \neq j$ which have identical pairs on at least one band are considered to be candidate pairs. In this case we would have below candidate pairs:

$\{ (C_1, C_4), (C_2, C_5), (C_1, C_6), (C_1, C_3), (C_4, C_7) \}$

Ans. 2

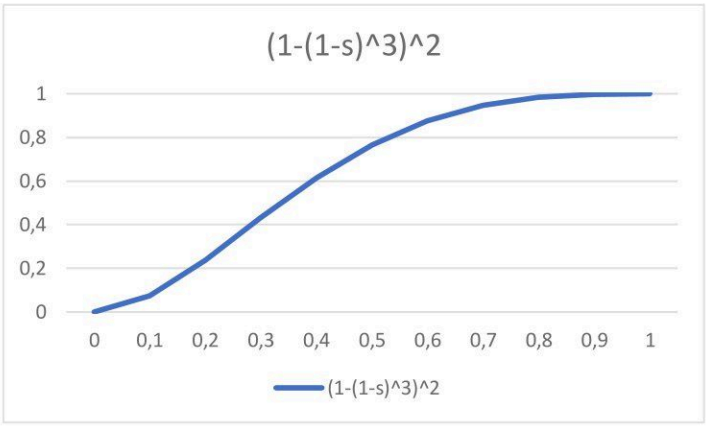
1)

s	1-(1-s^2)^3
0	0
0,1	0,029701
0,2	0,115264
0,3	0,246429
0,4	0,407296
0,5	0,578125
0,6	0,737856
0,7	0,867349
0,8	0,953344
0,9	0,993141
1	1



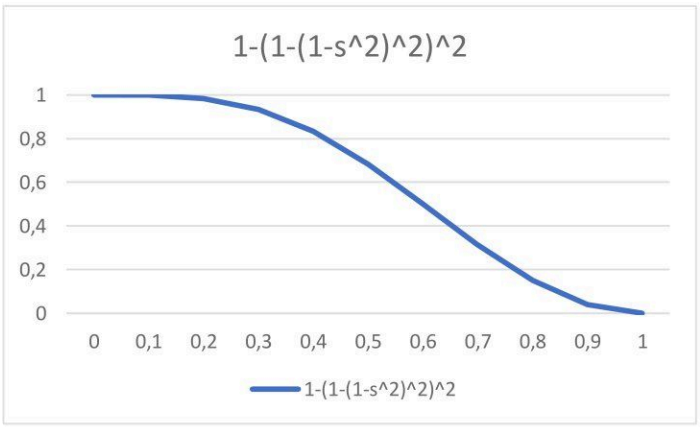
2)

s	(1-(1-s)^3)^2
0	0
0,1	0,073441
0,2	0,238144
0,3	0,431649
0,4	0,614656
0,5	0,765625
0,6	0,876096
0,7	0,946729
0,8	0,984064
0,9	0,998001
1	1



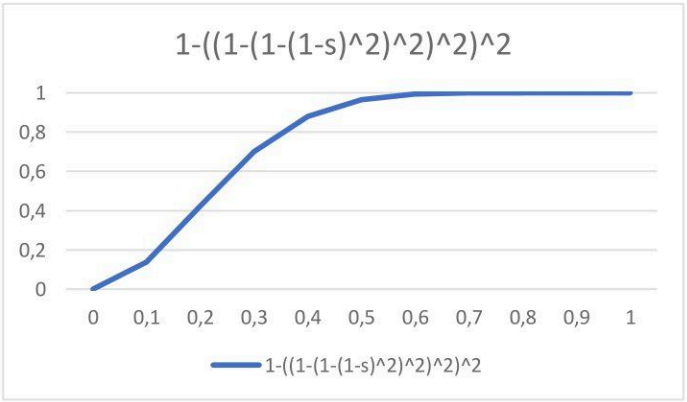
3)

s	1-(1-(1-s^2)^2)^2
0	1
0,1	0,99869679
0,2	0,98320384
0,3	0,93234799
0,4	0,83222784
0,5	0,68359375
0,6	0,50212864
0,7	0,31425039
0,8	0,15065344
0,9	0,03940399
1	0



4)

s	1-((1-(1-(1-s)^2)^2)^2)^2
0	0
0,1	0,136767225
0,2	0,426048058
0,3	0,700296297
0,4	0,878497449
0,5	0,963363647
0,6	0,992488075
0,7	0,999126821
0,8	0,99996222
0,9	0,999999843
1	1



After setting up snakemake environment — showing version and after creating Snakefile in the

```
(bda) pprince@ubuntuprince:~/Desktop/bda/snakemake-tutorial$ snakemake --version
6.15.5
(bda) pprince@ubuntuprince:~/Desktop/bda/snakemake-tutorial$ snakemake -n
Building DAG of jobs...
Nothing to be done (all requested files are present and up to date).
(bda) pprince@ubuntuprince:~/Desktop/bda/snakemake-tutorial$ snakemake -np
Building DAG of jobs...
Nothing to be done (all requested files are present and up to date).
(bda) pprince@ubuntuprince:~/Desktop/bda/snakemake-tutorial$ snakemake -n ()
bash: syntax error near unexpected token `('
```

directory running dry run with empty file.
Dry run newly created rule in snake file.

```
(bda) pprince@ubuntuprince:~/Desktop/bda/snakemake-tutorial$ snakemake -n --cores=2 course
sTaken-A.count coursesTaken-B.count
Building DAG of jobs...
Job stats:
job      count    min threads    max threads
-----  -------  -------------  -------------
count        1              1              1
total        1              1              1


[Sun Apr 30 15:12:54 2023]
rule count:
    input: coursesTaken-A.csv, coursesTaken-B.csv
    output: coursesTaken-A.count, coursesTaken-B.count
    jobid: 0
    resources: tmpdir=/tmp

Job stats:
job      count    min threads    max threads
-----  -------  -------------  -------------
count        1              1              1
total        1              1              1

This was a dry-run (flag -n). The order of jobs does not reflect the order of execution.
```

Running Snakefile without dry run.

```
(bda) pprince@ubuntuprince:~/Desktop/bda/snakemake-tutorial$ snakemake --cores=2 coursesTa
ken-A.count coursesTaken-B.count
Building DAG of jobs...
Using shell: /usr/bin/bash
Provided cores: 2
Rules claiming more threads will be scaled down.
Job stats:
job      count    min threads    max threads
-----  -------  -------------  -------------
count        1              1              1
total        1              1              1

Select jobs to execute...

[Sun Apr 30 15:14:13 2023]
rule count:
    input: coursesTaken-A.csv, coursesTaken-B.csv
    output: coursesTaken-A.count, coursesTaken-B.count
    jobid: 0
    resources: tmpdir=/tmp

[Sun Apr 30 15:14:35 2023]
Finished job 0.
1 of 1 steps (100%) done
Complete log: /home/pprince/Desktop/bda/snakemake-tutorial/.snakemake/log/2023-04-30T15141
3.781394.snakemake.log
```

Content of .count files

```
(bda) pprince@ubuntuprince:~/Desktop/bda/snakemake-tutorial$ cat coursesTaken-B.count
1000000(bda) pprince@ubuntuprince:~/Desktop/bda/snakemake-tutorial$ cat coursesTaken-A.count
1000000(bda) pprince@ubuntuprince:~/Desktop/bda/snakemake-tutorial$ snakemake --version
6.15.5
```

What did parameter —cores=2 do?
When the workflow is executed, the scheduler ensures that all jobs running at the same time does not exceed a given number of available CPU cores. This number is given with the —cores command line argument, which is mandatory for snakemake calls that actually run the workflow.
For ex: “snakemake —cores =2” will execute the workflow with 2 cores.