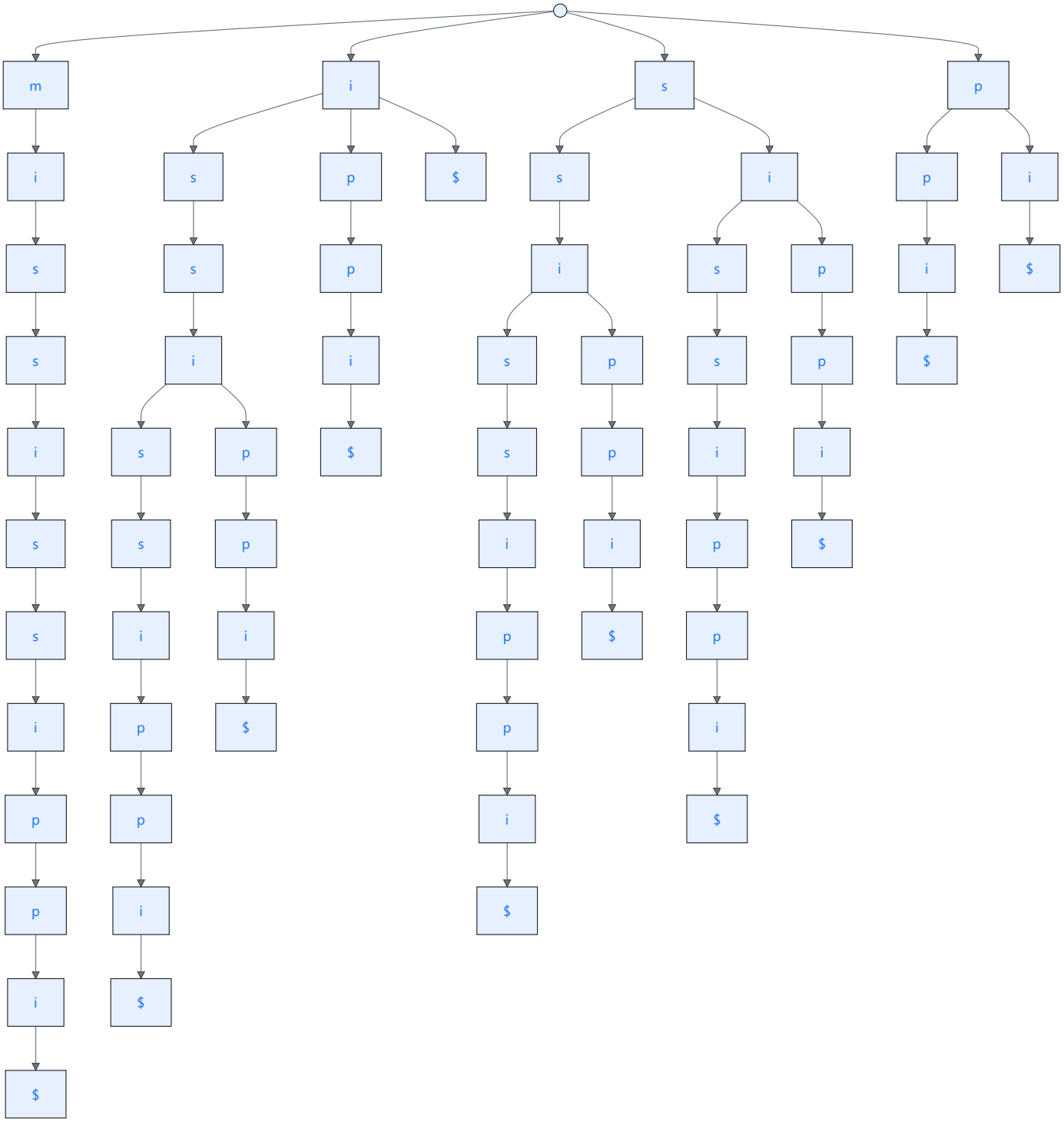
1. [Suffix trie] (30) Answer the following questions about suffix trie.

Construct a suffix trie of the string “mississippi”.

- i
- pi
- ppi
- ippi
- sippi
- ssippi
- issippi
- sissippi
- ssissippi
- ississippi
- mississippi



Consider a string “apple”, which of the suffixes below will be directly present in its suffix trie?

“ple”, “ap”, “e”, “apple”

> “ap” is not a suffix, the rest are and would be present in the suffix trie

Give True/False for the following statements:

Every leaf nodes in a suffix trie corresponds to a full suffix of the string.

> True

A suffix trie that contains the suffix “ba” must also contain “a”.

> True

A suffix trie always contains all the substrings of the string.

> False

2. [Inverted Index] (30) Consider three documents given below.

- Doc 1: “data retrieval is important”
- Doc 2: “data structures for retrieval”
- Doc 3: “efficient retrival techniques”

(a) (10) Following the example given in the slides, construct an inverted index of these three documents.

term	doc. freq	pointers	postings list
data	2	$\rightarrow$	$1 \rightarrow 2$
effecient	1	$\rightarrow$	3
for	1	$\rightarrow$	2
important	1	$\rightarrow$	1
is	1	$\rightarrow$	1
retrieval	3	$\rightarrow$	$1 \rightarrow 2 \rightarrow 3$
structures	1	$\rightarrow$	2
techniques	1	$\rightarrow$	3

(b) (20) Answer the following questions.

What change needs to be made to the inverted index, if we remove “is” and “for”?

> We remove those rows from the inverted index

What happens to the inverted index, if a fourth document is added with the following text: “retrieval techniques is fast”?

> Add new row (fast, 1, $\rightarrow$, 4)
>
> Edit retrieval and techniques rows to have document 4
>
> If we take the above change into account, readd a row (is, 1, $\rightarrow$, 1)
>
> Otherwise we edit “is” row to have document 4

Describe how the inverted index is used to find the answer for a keyword query “retrival AND data” and give the answer.

> Locate retrival in dictionary, retrieve postings ($1 \rightarrow 2 \rightarrow 3$).
>
> Locate data in dictionary, retrieve postings ($1 \rightarrow 2$.
>
> Intersect two posting sets, resulting in document 1, 2

Explain how you will modify your inverted index to support “phrase searches”, for the following phrases: “data retrieval”, “data structures”, and “retrival techniques”.

> In the postings store for each term the position in which it appears in each document
>
> When doing the phrase search, for each term in the phrase extract the inverted index entries.
>
> Then merge their doc:position lists to enumerate all positions with the query

3. Query extension for Data Streams A weather station generates a stream of measurements with schema: Sensors (sensorID, temperature, time), where each tuple records the temperature reading of a sensor at a timestamp. SELECT-FROM-WHERE SQL queries can be extended by adding a sliding window to search over data streams. For example,

- Sensors [Rows 2000] describes a window on the Sensors stream consisting of the most recent 2000 tuples (or records);
- Sensors [Range 15 seconds] describes a window on the Sensors stream consisting of the tuples that arrived within the last 15 seconds.

For example, a query SELECT Max(temperature) FROM Sensors [Range 1 hour] expresses "What's the maximum temperature recorded in the last hour?".

(a) (30) Give Select-from-where queries for the following questions:

Which sensors capture a temperature above 50 degrees in the last 5 hours? Find their ids.

```
SELECT sensorID FROM Sensors [RANGE 5 hour] WHERE temperature > 50;
```

What is the maximum temperature recorded by sensors with id = '102'?

```
SELECT MAX(temperature) FROM Sensors [Range All] WHERE sensorID = '102';
```

What is the average temperature recorded by sensors with ids in range [100,110], in the most recent 1500 records?

```
SELECT AVG(temperature) FROM Sensors [Rows 1500] WHERE sensorID BETWEEN 100 AND 110;
```

Assume every sensor re-detect and updates records every hour. Write a query that finds the average temperature in the past 24-hous –without using "Range 24 hour".

```
SELECT AVG(temperature) FROM Sensors [Rows 24]
WHERE sensorID IN (SELECT DISTINCT sensorID FROM Sensors [Rows 24]);
```

Given a query Q:

SELECT Min(temperature) FROM Sensors [Rows 6] WHERE Sensor.sensorID = '100' OR Sensor.sensorID = '102'

and a stream S of sensor readings with all the tuples generated by all the sensors:

S = (100, 80, 0), (100, 70, 10), (102, 80, 10), (103, 75, 15), (100, 82, 20), (100, 80, 25), (100, 75, 30), (102, 65, 35), (103, 80, 40), (100, 70, 50)

What is the most recent output of the query Q over S?

> The last most recent 6 rows for sensors 100/102 are:
>
> (100,70,50), (100,75,30), (102,65,35), (100,82,20), (100,80,25), (102,80,10)
>
> So the minimum temperature is 65 which is the output of the query

(b) (10) Consider a second stream of measurements with schema: Sensor2 (sensorID, nitrogen, BSI, time). A SQL statement can be extended to join the two streams on common attributes. For example, the query below integrates temperature and nitrogen monitored by the same sensors.

SELECT * FROM Sensor1 JOIN Sensor2 WHERE Sensor1.sensorID = Sensor2.sensorID

Harmful algal blooms (HAB) occurs when colonies of algae grow out of control and produce toxic or harmful effects on people, fish, birds and marine mammals. This is evaluated by BSI (Bloom Severity). A "Do not drink" alert was issued due to a 2014 HAB event on Lake Erie. Assume now sensors with ids 1 – 500 are deployed for Lake Erie. write SQL-like queries to monitor HAB status of Lake Erie by answering the following two questions.

- Assume you can use arbitrarily long sliding window for your queries. What is the average temperature of the areas of Lake Erie where any sensor has detected level two Bloom Severity (BSI > 2), in the past 72 hours?

```
SELECT AVG(S1.temperature) FROM Sensor1 [Range 72 hour] S1
JOIN Sensor2 [Range 72 hour] S2 ON S1.sensorID = S2.sensorID
WHERE S2.BSI > 2 AND S1.sensorID BETWEEN 1 AND 500;
```

- Assume you can have SQL queries with sliding window with a length one week at interval every 1 hour. But you can store and archieve the monitored results in a new table. Write queries to find out the answer for the following question:

What is the range of the nitrogen readings in the areas of Lake Erie where a level three Bloom Severity is detected in all the historical data from 2020 – 2023?

```
SELECT MIN(S2.nitrogen), MAX(S2.nitrogen) FROM Sensor2 [Range All] S2
WHERE S2.BSI = 3 AND S2.sensorID BETWEEN 1 AND 500
AND S2.time BETWEEN '2020-01-01' AND '2023-12-31';
```