CSC 370 - SUMMER 2020
DATABASE SYSTEMS
ASSIGNMENT 5
UNIVERSITY OF VICTORIA

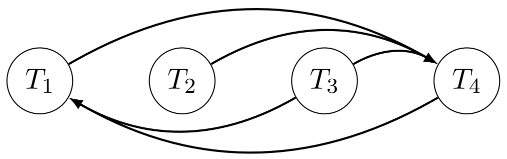
Due: Friday, July 24th, 2020 at **18:00 Victoria Time (PDT)**. **Late assignments will not be accepted.**

Answers must be well formatted and legible or they will not be marked.

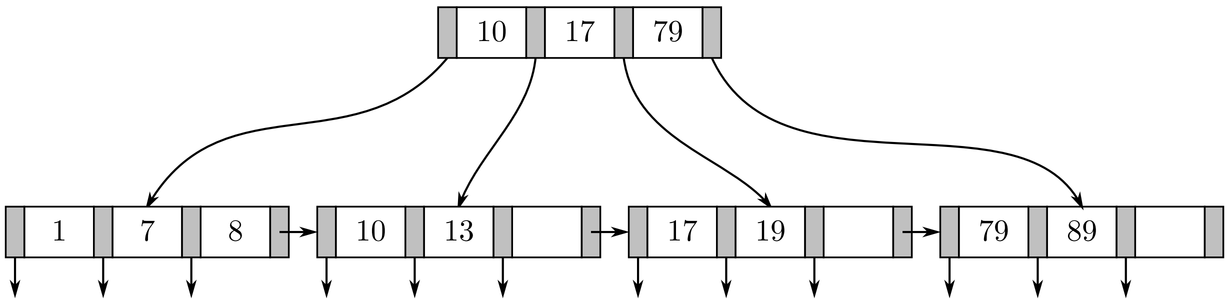
Question 1: Conflict-Serializability [6 marks]
Each part below contains a schedule of read and write operations on various data elements by various transactions. For each part, determine whether the schedule is conflict-serializable. If the schedule is conflict-serializable, provide a serial schedule that is conflict-equivalent.

(a) $S = R_1(X), R_3(Y), W_2(X), R_2(X), W_2(Y), W_1(Y), W_3(Y), W_1(X), R_1(Y)$
(b) $S = W_2(X), R_1(X), W_2(Y), W_1(Y), R_3(Y), R_2(X), W_3(Y), W_1(X)$
(c) $S = R_1(Y), R_3(X), W_2(Y), R_2(Y), W_2(X), R_1(Y), W_3(X), W_1(Y)$
(d) $S = R_3(X), R_2(X), R_1(X), R_3(Y), W_1(X), R_2(Y), R_1(X), R_2(Y), W_2(Y)$

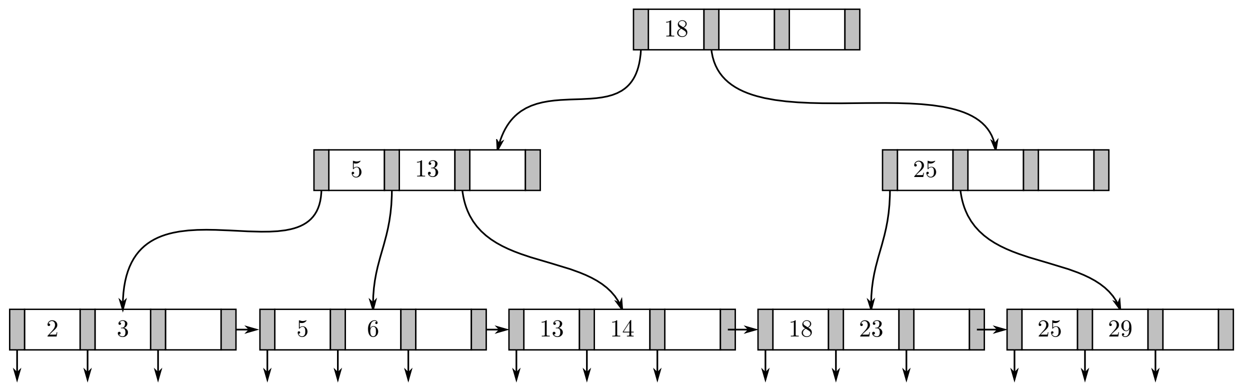
Question 2: Precedence Graphs [2 marks]
Design a schedule of read (R) and write (W) operations on three data items X, Y and Z such that the result has the following precedence graph. For full marks your answer must use as few operations as possible.



Question 3: B+ Tree Insertion [2 marks]
Draw the B+ Tree produced after inserting the value 6 into the B+ Tree below.



Question 4: B+ Tree Removal [2 marks]
Draw the B+ Tree produced after deleting the value 6 from the B+ Tree below. Remember to enforce the density constraints.



Question 5: Indices [4 marks]
Recall from Assignment 4 that the `observations` table in the `vwsn_1year` database contains a huge amount of data, and operations on the whole table can be very time consuming. The `CREATE` statement for the `observations` table is given below.

```
create table observations(
        station_id int,
        observation_time timestamp,
        temperature real,
        primary key(observation_time, station_id),
        foreign key(station_id) references stations(id) on delete cascade on update cascade
        );
```

Suppose that `observations` contains 10,000,000 rows divided into 80,000 blocks, with every index implemented with a B+ Tree[1]. Each B+ Tree node requires one disk block and stores up to 250 keys. Assume that all data is stored strictly on disk (so any access will not be affected by data cached to main memory).

(a) Suppose that a sparse clustering index is defined on the primary key (`observation_time`,`station_id`). How many disk block read operations are needed to find and retrieve a single observation row given a time and station ID in the worst case?

(b) Suppose that a dense non-clustering index is defined on `temperature`. If a query is used to find all observations with `temperature` = 10, and there are 40,000 such rows in the dataset, how many disk block read operations will be needed in total (including traversing any indices and actually reading the rows) in the worst case?

1. These numbers are based on the actual storage requirements of the table as reported by the database server, but rounded to more convenient values.