CPT201 Coursework 1

Start date:	*16:00, 20 Sep 2024*
Deadline:	*16:00, 27 Sep 2024*
Learning outcomes assessed:	*A*
Percentage in final mark:	*15%*
Late submission policy:	*5% of the total marks available for the assessment shall be deducted from the assessment mark for each working day after the submission date, up to a maximum of five working days.*
Risk:	• *Please read the coursework instructions and requirements carefully. Not following these instructions and requirements may result in lose of marks.* • *Plagiarism results in award of ZERO mark; all submissions will be checked by TurnItIn.* • *The formal procedure for submitting coursework at XJTLU is strictly followed. Submission link on Learning Mall will be provided in due course. The submission timestamp on Learning Mall will be used to check late submission.*

Description

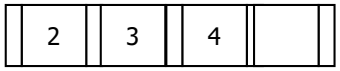
- Coursework 1 is about the update algorithms for B+ tree and extendable hash indices.
- ***You MUST following the B+tree insertion and deletion algorithms in the textbook or lecture.***
- It accounts for 15% of the final assessment.
- CW1 questions:
  - Question 1 is on B+ tree index, 40 marks.
  - Question 2 is on extendable hash indices, 30 marks.
  - Question 3 is a case study, 30 marks.

Important Notes

- Marking
  - Marks will be awarded for each correct step.
  - By a correct step, it means everything in that step is correct.
  - If a step is wrong, then it is likely all the subsequent steps will be wrong.
  - The module leader reserves the right to interview students should they be suspected for plagiarism. Not attending the interview will be reported to the department and school exam officers.
- Submission
  - Prepare your answers based on the ***answer template*** in Powerpoint (will be provided).
  - One index/diagram per slide.
  - You must submit a ***PDF*** file. Submit via the link on Learning Mall Core. The system is configured to accept ONLY PDF file.

Q1 B+ Tree Update (20*2 marks) = 40 marks

- B+ tree, n=5 (i.e. 5 pointers in one node).
- Based on the initial tree below, draw B+ tree indices after the following updates. In total, there should be 20 B+ tree indices. Every update must be performed based on the previous tree.
  - Insert search keys [11, 7, 18, 37, 19, 8, 77, 80, 28, 79, 88, 4, 6, 44, 66, 98], in the exact order.
  - Delete search keys [19, 21, 88, 3], in the exact order.
- Initial tree



Q2 Extendable Hash Index Construction (15*2 marks) = 30 marks

- Assume that the initial hash index is empty, and one block can hold up to FOUR records. Based on the city names and their hash values (already computed based on the hash function) in the following table, draw hash indices after inserting the following 15 records, in the exact order. You should have 15 indices in total. Every insertion must be performed based on the previous hash index.
- In insertion, overflow buckets should be used if after *TWO* splits have been tried and insertion is still unsuccessful (revert back to the original index).
- Records to be inserted, in the exact order:
  - Taiyuan, Cambridge, Kuala Lumpur, Florence, Lianyungang, Nottingham, Taiyuan, Cambridge, Venice, Kuala Lumpur, Java Island, Valencia, Florence, Chiang Mai, Nottingham.

Q2 Extendable Hash Index Construction - cont'd

city name	hash value: h(city name)
Lianyungang	1101 0111 1011 0101 1110 0101 1110 1111
Nottingham	0010 1010 1101 0101 0111 0010 1100 0101
Cambridge	0110 1001 1000 0011 1100 0010 0101 0000
Venice	0011 0001 0011 0100 1101 1101 0111 0101
Kuala Lumpur	1000 0001 1100 0000 1010 0001 1100 1110
Taiyuan	0010 1001 1000 0011 1100 0010 0101 0000
Java Island	0111 1001 1000 0011 0110 1001 1000 0011
Valencia	1110 1001 1000 0011 1100 0010 0101 0000
Florence	0010 1001 1000 0011 0110 1001 1000 0011
Chiang Mai	0010 1110 0001 0001 0101 0111 1100 1100

Q3 Case Study (10*3 marks) = 30 marks

- XJTLU is building a medium-sized relational database to store its academic publication data. The envisaged system is to support effective and efficient retrieval of staff members' publications, similar to Google Scholar (e.g. search by year, author name, title or abstract). Only B+ tree and extendable hash indexing techniques are considered. An example of a publication record in bibtex is shown in the figure below.
- Questions
  - (1) If an index to be built on year/month, briefly discuss which indexing technique is preferred. Maximum 50 words.
  - (2) If an index to be built on author names (names can be separately indexed), briefly discuss which indexing technique is preferred. Maximum 50 words.
  - (3) To support keyword-based full text search, If an index to be built on abstract, briefly discuss which indexing technique is preferred. Maximum 50 words.

Q3 Case Study – cont'd

```
@inproceedings{10.5555/3295222.3295349,
        author = {Vaswani, Ashish and Shazeer, Noam and Parmar,
Niki and Uszkoreit, Jakob and Jones, Llion and Gomez, Aidan N. and
Kaiser, \L{}ukasz and Polosukhin, Illia},
        title = {Attention is all you need},
        year = {2017},
        abstract = {The dominant sequence transduction models are
based on complex recurrent or convolutional neural networks that
include an encoder and a decoder. The best performing models also
connect the encoder and decoder through an attention mechanism. We
propose a new simple network architecture, the Transformer, based
solely on attention mechanisms, dispensing with recurrence and
convolutions entirely. Experiments on two machine translation tasks
show these models to be superior in quality while being more
parallelizable and requiring significantly less time to train. Our
model achieves 28.4 BLEU on the WMT 2014 English-to-German
translation task, improving over the existing best results,
including ensembles, by over 2 BLEU. On the WMT 2014 English-to-
French translation task, our model establishes a new single-model
state-of-the-art BLEU score of 41.0 after training for 3.5 days on
eight GPUs, a small fraction of the training costs of the best
models from the literature.},
        booktitle = {Proceedings of the 31st International
Conference on Neural Information Processing Systems},
        pages = {6000–6010},
}
```