COMP PhD Qualifier Sample Exam

- Time: 09:30 - 12:30
- By submitting this exam, you agree to fully comply with Koç University Student Code of Conduct, and accept any punishment in case of failure to comply. If you do not submit this exam on time, you will directly fail the qualifier.
- In general, you may neither use other resources during the exam, nor ask any questions.
- Good luck!

1. [**agursoy 17 pts**] Two questions:

   (a) Let T be a red-black tree storing n items, and let k be the key of an item in T. Show how to construct from T, in O(log n) time, two red-black trees T' and T", such that T' contains all the keys of T less than k, and T" contains all the keys of greater than k. This operation destroys T.

   (b) Consider the following abstract data type (ADT)

   ```
   ADT Queue
       add(x):  insert x into collection
       delete():  remove the oldest element, and return it
       create(n):  create an empty queue of maximum size n
   ```

   Implement Queue ADT using arrays in your favorite language. The add and delete operations should run in O(1) time.

2. [**dyuret 17 pts**] We are given three coins. The first coin is a fair coin painted blue on the head side and white on the tail side. The other two coins are biased so that the probability of a head is p. They are painted blue on the tail side and red on the head side. Two of the three coins are to be selected at random and tossed.

   - Draw the sample space.
   - What is the probability that the sides that land face up are of the same color (in terms of p)?
   - What are the possible values of p if the probability that the sides that land face up are of the same color is 29/96?

3. [**yyemez 17 pts**] Show that graph isomorphism is an equivalence class.

4. [**oozkasap 17 pts**]

   (a) Given the reference string 5, 3, 4, 3, 1, 2, 1, 4, 3, 4, 5, 3, 2 which indicates the sequence of page accesses by a running program, you are asked to develop the following tables for page replacement algorithms.

   i. With three (initially empty) page frames, how many page faults will occur with LRU (Least Recently Used) page replacement? Show the contents of the page frames at each step, circling newly loaded pages.

5	3	4	3	1	2	1	4	3	4	5	3	2

   ii. With three (initially empty) page frames, how many page faults will occur with FIFO (First in First out) page replacement? Show the contents of the page frames at each step, circling newly loaded pages.

5	3	4	3	1	2	1	4	3	4	5	3	2

   iii. Explain what Belady's anomaly is, and which page replacement algorithms mentioned above are subject to Belady's anomaly?

   (b) Consider a page reference string for a process with m frames in main memory (initially all empty). The page reference string has length k (i.e. k page references) with s distinct page numbers occurring in it. For any page replacement algorithm:

   i. What is the lower bound on the number of page faults? Explain briefly.

   ii. What is the upper bound on the number of page faults? Explain briefly.

5. [**akupcu 17 pts**] Recurrence

```
Spiral (n)
if n <= 1 then return
drive north n miles
drive west n miles
drive south n miles
drive east n-1 miles
drive north 1 mile
run Spiral (n-2)
```

Let M(n) be the total number of miles driven when Spiral(n) is called.

(a) M(1) = ?

(b) M(2) = ?

(c) M(n) = ? recursively

(d) M(n) = O(?)

6. [**stasiran 17 pts**] Consider an object-oriented language such as C++, C# or Java and its collections library that provides data structures such as lists, sets and hash maps. Suppose that you are given two sorted lists of E objects called list1 and list2. Assume that, for E objects e1 and e2, e1.compareTo(e2) returns 0 if the two objects are equal, a negative integer if the object e2 is ”smaller” (should come earlier in the sorted list) than the object e1, and a positive integer otherwise.

   Write a method that creates and returns a new, sorted list list3 that is obtained by merging elements from list1 and list2. What is the complexity (in terms of the sizes of list1 and list2, assuming compareTo() is constant time) of your solution?

7. [**eyilmaz 17 pts**] Query optimizers implemented in many relational database management systems take a given SQL query, convert it into an equivalent representation in relational algebra, and use equivalences between different relational algebra expressions to convert the initial query into one that can be executed faster.

   In the next five questions, you are given two relational algebra expressions separated by an equivalence sign (≡). For each question, write **True** if the expressions are equivalent and **False** if they are not and explain why. If the expressions are not equivalent, state a condition under which they become equivalent.

(i) $\sigma_C(\pi_{A_1,A_2,\ldots A_n}(R)) \equiv \pi_{A_1,A_2,\ldots A_n}(\sigma_C(R))$

(ii) $\sigma_C(R \times S) \equiv \sigma_C(R) \times S$

(iii) $\pi_{A_1,A_2,\ldots A_n}(R - S) \equiv \pi_{A_1,A_2,\ldots A_n}(R) - \pi_{A_1,A_2,\ldots A_n}(S)$

(iv) $\pi_{A_1,A_2,\ldots A_n}(R \cup S) \equiv \pi_{A_1,A_2,\ldots A_n}(R) \cup \pi_{A_1,A_2,\ldots A_n}(S)$

(v) $\sigma_{C \texttt{ AND } D}(R) \equiv \sigma_C(\sigma_D(R))$

8. [**msezgin 17 pts**] Consider the following piece of program:

```
let x = 5
    in let y = 2
       in let f = proc(x) -(x,-(0,x))
          in let g = proc(x) begin set x=15; y end
             in -( (f begin set x=-(x,-(0,x)); x end), (g (g y)) )
```

(a) What is the value of the program under call-by-value?

(b) What is the value of the program under call-by-reference?

(c) What is the value of the program under call-by-need?

(d) What is the value of the program under call-by-name?