Computer Science B63
University of Toronto Scarborough

Homework Assignment # 4

---

**Handing in and marking**

For this assignment, you need to submit your solutions to the pencil-and-paper exercises on crowdmark and your solutions to the programming question on MarkUs. Your pencil-and-paper solutions will be marked with respect to correctness, clarity, brevity, and readability. Your code will be marked with respect to correctness, efficiency, program design and coding style, clarity, and readability. This assignment counts for 10% of the course grade.

---

## Question 1. Disjoint Sets [25 MARKS]

Consider the following sequence of operations:

```
for i in 1,2, ..., 16: make-set(i)
for i in 1,3, ..., 15: union(i, i+1)
for i in 1, 5, 9, 13: union(i, i+2)
union(1,5); union(11,13); union(1,10)
find-set(2)
find-set(16)
```

Show the resulting data structure, if disjoint sets are implemented using:

1. linked lists (as we did in class, shorter list merged into longer list), assuming that if set $(i)$ and set $(j)$ have the same size, the algorithm `union(i, j)` appends elements of $(j)$ to elements of $(i)$,
2. disjoint-set forests with union by rank, but without path compression, and
3. disjoint-set forests with union by rank and with path compression.
4. Prove, by strong induction on the number of nodes, that every node in the forst implementation of disjoint sets has rank at most $\lfloor \lg n \rfloor$, where $n$ is the number of nodes in the set.

## Question 2. Average Running Time [20 MARKS]

1. Consider the following function:

   ```
   while True:
       print("Hello World")
       i := random(1, n)
       if i <= k:
           break
   ```

   How many times would you expect this function to print "Hello World".
2. Write the function from part 1 in any language of your choice. Run the function 10,000 times and count the number of "Hello World" that get printed. Create a bar plot for frequency of number of "Hello World" prints you observe (x-axis will have count of prints observed, y-axis will have the frequency). Choose any reasonable value for $n$ and $k$ (static across all runs). Comment if your expectation from part 1 is also observed practically.
3. Consider a list of $n$ positive integers $A$ and the following algorithm:

   ```
   best = -1
   for every i in A:
       if i > best:
           process(i)
           best = i
   ```

If the function `process(i)` takes runtime cost of $C_p$. What is the best case, worst case and average case running time of the entire algorithm.

## Question 3. Hashing [15 MARKS]

1. Consider a hash table with 9 slots and the hash function $h(k) = k \bmod 9$. Demonstrate what happens upon inserting the keys 5, 28, 19, 15, 20, 33, 12, 17, 10 with collisions resolved by chaining (closed addressing).
2. Consider inserting the keys 10, 22, 31, 4, 15, 28, 17, 88, 59 into a hash table of length $m = 11$ using open addressing. Illustrate the result of inserting these keys using linear probing with $h(k, i) = (k + i) \bmod m$.
3. Perform the same as part (2) but instead of using linear probing, use double hashing with $h_1(k) = k$ and $h_2(k) = 1 + (k \bmod (m - 1))$.

## Question 4. B-Trees [20 MARKS]

1. Starting from an empty B-tree insert the following keys in the B-tree of fanout 5. Show the tree after every insert.

   F, S, Q, K, C, L, H, T, V, W, M, R, N, P, A, B, X, Y, D, Z, E
2. In the above tree, perform deletion of key F, B, C, S, R in the given order. Show the tree after every delete.