Midterm Exam: Analysis of Algorithms
Dr. Mudassir Shabbir, March 19, 2025

The exam consists of a total of 100 points, and you have 180 minutes. Make sure to write your roll-number and name on the question paper AND answersheet. **The solution to every question must start on a new page and complete answer to each question (all parts!) on a single page. Anything written beyond first ten pages of your answersheet will not be graded!**

---

1 Priority Queues

1. Give an $O(n \log k)$-time algorithm to merge k sorted lists into one sorted list, where n is the total number of elements in all the input lists. (Hint: Use a min-heap for k-way merging.)
2. Where in a max-heap might the smallest element reside, assuming that all elements are distinct?

2 Union-Find

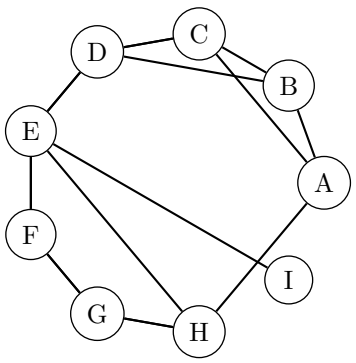
1. Prove that the height of a union-find data structure with path compression is at most $\log n$, i.e., $H(UF) \leq \log n$.
2. Consider the following sequence of Union-Find operations on elements {1,2,3,4,5,6}. Show the forest structure after each operation, assuming both path compression and union by rank are used. In the start, all elements are in their own set.
   (a) Union(1,2)
   (b) Union(3,4)
   (c) Union(5,6)
   (d) Find(4)
   (e) Union(2,3)
   (f) Find(6)
   (g) Union(1,5)
   (h) Find(6)

3 Let us consider the random experiment of rolling two dice.

1. Define a random variable X as the number of 6's that you get. Find the expected value of X.
2. Define a random variable Y as the sum of the two dice. Find the expected value of Y.

4 Articulation Points

1. Similar to articulation points, a *bridge* (or cut edge) is an edge whose removal increases the number of connected components. Provide an $O(V + E)$ algorithm to identify all bridges in an undirected graph. Show how the DFS tree can be used to detect these edges.
2. For the following undirected graph on eight vertices, find all articulation points using the algorithm discussed in the class. Show the discovery time and low value for the whole graph each time an articulation point is found.



5 Suppose you are given a "black box" linear-time median finding algorithm. Design a simple linear-time algorithm to find k^{th} smallest element using this algorithm. Note that median finding algorithm that you are given is black-box and you can't change it in anyway; you can just call it to find median of a list in linear-time.

6 How can the number of strongly connected components of a graph on n vertices change if a new edge is added? Give the extreme cases where it changes drastically.

7 Counting sort can also work efficiently if the input values have fractional parts, but the number of digits in the fractional part is small. Suppose that you are given n numbers in the range 0 to k, each with at most d decimal (base 10) digits to the right of the decimal point. Modify counting sort to run in $\Theta(n + 10^d k)$ time.

8 You are given a **dataset containing 500GB** of unsorted **transaction records**, but your system has only **8GB of RAM** available for processing. Since the dataset is too large to fit into memory at once, a straightforward in-memory sorting approach is not feasible. Devise a plan to **efficiently sort** this dataset while minimizing I/O operations.

9 The Strassen matrix multiplication algorithm follows the recurrence $T(n) = 7T(n/2) + O(n^2)$. Use the Master Theorem to determine the time complexity of this algorithm.

10 Give runtime complexity of the following algorithm.

```
procedure ALGO1(x)                                   ▷ input x is a number
    if |x| > 2 then
        return ALGO1(√x)
    else
        return x
    end if
end procedure
```