Due 23:59 Feb 25 (Sunday). There are 100 points in this assignment. Submit your answers (**must be typed**) in pdf file to CourSys
`https://coursys.sfu.ca/2024sp-cmpt-307-d1/`.
Submissions received after 23:59 will get penalty of reducing points: 20 and 50 points deductions for submissions received at $[00:00, 00:10]$ and $(00:10, 00:30]$ of Feb 26, respectively; no points will be given to submissions after $00:30$ of Feb 26.

1. 10 points (Ex 11.2-1 of text book)

   A hash function h hashes n distinct keys into an array T of m elements. Assuming simple uniform hashing, what is the expected number of collisions, that is, the expected value of $|\{\{k,l\}|k \neq l \text{ and } h(k) = h(l)\}|$.

2. 10 points (Ex 11.2-2 11.3-4 of text book)

   (a) Hash function $h(k) = k \bmod 9$ is used to insert keys $5, 28, 19, 15, 20, 33, 12, 17, 10$ into a hash table $T[0..8]$ with collisions resolved by chaining. Show the result of the hash table.

   (b) Hash function $h(k) = \lfloor m(kA - \lfloor kA \rfloor) \rfloor$ for $A = (\sqrt{5} - 1)/2$ is used to insert keys $61, 62, 63, 64, 65$ into a hash table $T[0..999]$. Show the locations of the hash table to which these keys are mapped.

3. 20 points (P 11-4 of text book)

   A class $\mathcal{H}$ of hash functions which map the universe U of keys to $\{0, 1, .., m-1\}$ is k-independent if, for every fixed sequence of k distinct keys $(x_1, .., x_k)$ and for any hash function h chosen at random from $\mathcal{H}$, the sequence $(h(x_1), .., h(x_k))$ is equally likely to be any of the m^k sequences of length k with elements drawn from $\{0, 1, .., m-1\}$.

   (a) Show that if $\mathcal{H}$ is 2-independent then $\mathcal{H}$ is universal.

   (d) Assume that Alice and Bob secretly agree on a hash function h from a 2-independent class $\mathcal{H}$ of hash functions. Each $h \in \mathcal{H}$ maps the keys in a universe U to $\mathbb{Z}_p$, where p is prime. Alice sends (x, t), where $x \in U$ and $t = h(x)$ is an authentication tag, to Bob over the Internet. Bob checks the pair (x, t) he receives indeed satisfies $t = h(x)$. Assume that an adversary intercepts (x, t) en route and tries to fool Bob by replacing the pair (x, t) with a different pair (x', t') with $x' \in U$, $x' \neq x$, $t' = h'(x')$ and $h' \in \mathcal{H}$. Prove the probability that the adversary succeeds in fooling Bob into accepting (x', t') is at most $1/p$.

4. 10 points (Ex 12.1-5)

   Prove that in the worst case in the comparison model, any comparison-based algorithm takes $\Omega(n \log n)$ time to construct a binary search tree from an arbitrary list of n elements. (Hint: apply the $\Omega(n \log n)$ lower bound on the worst case running time of comparison-based sorting algorithm to sort n numbers.)

5. 15 points

   Implement the algorithms Cut-Rod(p, n) (slide page 4), Memoized-Cut-Rod(p, n) (slide page 6) and Bottom-Up-Rod(p, n) (slide page 7) discussed in class for the rod-cutting problem by a same programming language (any language is OK) on a same computing plaform. Report the running times of the three algorithms on a computer for $n = 5, 10, 15, 20, 25, 30$, respectively. For each instance size n, use a same price table p to run the three algorithms. Submission of codes is not needed.

6. 15 points (Ex 15.1-3 of text book)

   A modified rod-cutting problem is that, in addition to a price p_i for each rod, each cut incurs a fixed cost of c, and the revenue associated with a solution is the sum of the prices of the pieces minus the costs of making the cuts. Give a dynamic-programming algorithm (optimal solution structure, Bellman equation, pseudo code and running time) to find the maximum revenue of the modified problem.

7. 20 points

   A digraph G with nodes $v_1, .., v_n$ is an ordered graph if it has the following properties: (i) Every directed edge has the form (v_i, v_j) with $i < j$ and (ii) for every node v_i, $i = 1, 2, .., n-1$, there is at least one edge of the form (v_i, v_j). The length of a path in G is the number of edges in it. Given an ordered digraph G, the goal is to find the length of the longest path that begins at v_1 and ends at v_n. Give a dynamic programming algorithm (optimal solution structure, Bellman equation, pseudo code and running time) which, given an ordered graph, finds the length of the longest path that begins at v_1 and ends at v_n.