Short answers

1. True or False? Circle your answers. No justification. **Wrong answers will receive a penalty of -1**.

   (a) (2 points) Performing one rotation always preserves the AVL property.
   A. True **B. False**

   (b) (2 points) In a r[illegible]st half of the nodes in a path from the root to a leaf are red.
   A. True B. [illegible]

   (c) (2 points) The [illegible]visited by the depth-first search algorithm (DFS) algorithm is always strictl[illegible]der of the vertices returned by the breadth-first search algorithm (BFS) alg[illegible]
   A. True **B. F[illegible]**

   (d) (2 points) Let $P(n)$ be a property over a variable n. We want to prove by induction that $P(n)$ is true for all $n \geq n_0$. Assume the base case $P(n_0)$ is true. Then, for the inductive case, only assuming that $P(n-1)$ is true is always sufficient to prove that $P(n)$ is true too.
   A. True **B. False**

   (e) (2 points) If $f(n)$ is $O(g(n))$ then $g(n)$ is $O(f(n))$.
   A. True **B. False**

   (f) (2 points) We implement hash tables using the open addressing technique to solve conflicts. In this implementation, the load factor α cannot exceed 1.
   A. True B. False

   (g) (2 points) We run the depth-first search algorithm (DFS) on a graph G and identify a back edge. Thus, G has at least one cycle.
   A. True B. False

   (h) (2 points) Given a partition of the vertices of a weighted undirected graph, the cut has one and only one light edge.
   A. True **B. False**

   (i) (2 points) We run the Dijkstra's algorithm on a graph with a negative-weight cycle. Then, the algorithm could not terminate.
   A. True **B. False**

   (j) (2 points) A bipartite graph has no cycle.
   A. True **B. False**

Multiple choices

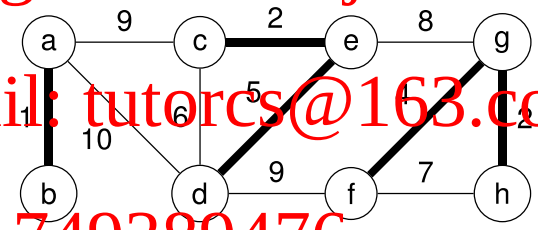
2. Multiple choice questions. No justification. No negative penalty for wrong answers. A question has at least one correct answer but can also have multiple correct answers. You will receive full credits if and only if you identify all correct answers and do not include wrong ones.

   (a) (5 points) We w…) $= \frac{1}{2} \cdot n^2 + 8 \cdot \sqrt{n} + 1$ is $O(n^2)$ using the formal definition introduced in cl…$(n))$ if there are two constants c and n_0 such that $f(n) \leq c \cdot g(n)$ for all n …ollowing pairs(s) of constants c and n_0 satisfy the definition?

   A. $c = \frac{2}{2}$
   B. $c = \frac{1}{2}$
   C. $c = \frac{1}{2}$
   D. $c = 10$ **and** $n_0 = 1000$

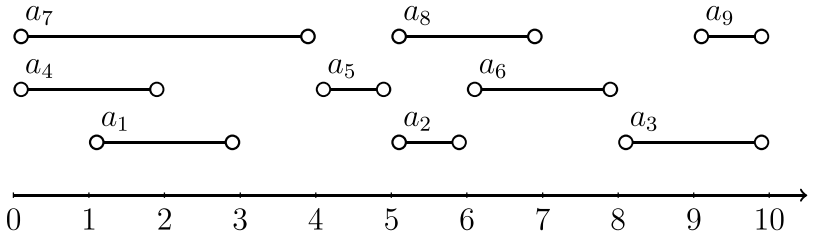
   (b) (5 points) We are in the middle of the execution of the Kruskal's algorithm for computing a minimal spanning tree (MST) of the graph below. The bold edges have already been selected to be in the solution found so far by the Kruskal algorithm, and the edge (d, e) has just been added to the MST (i.e. under construction). Which edge will be the next one to be added to the MST?

   A. (e, g) B. (e, d) C. (a, f) D. (f, h)



   (c) (5 points) What is/are the longest series of activities (i.e. number of activities) that could be returned by a greedy algorithm for solving the scheduling problem (i.e. finding the maximal number of compatible activities) as seen in class?

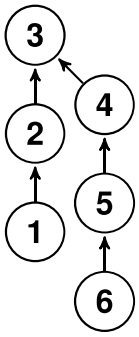
   A. $(a_7, a_5, a_2, a_6, a_3)$ B. $(a_7, a_5, a_2, a_6, a_9)$ **C.** $(a_4, a_5, a_2, a_6, a_3)$ **D.** $(a_4, a_5, a_2, a_6, a_9)$



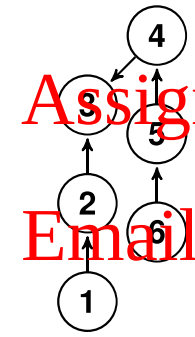
(d) (5 points) We use a tree representation to model disjoint sets, and implement unions as *union-by-height* with path compression. We have the two following trees T_1 and T_2 representing two distinct disjoint sets.

(B) T_2

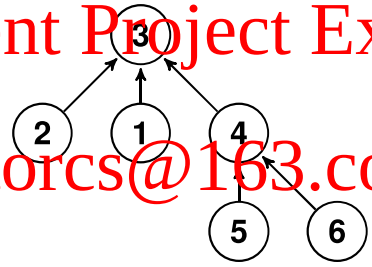
We perform the union of the set containing 6 with the set containing 1 (i.e. `union(6,1)`). Which of the options below is/are possible output(s)?



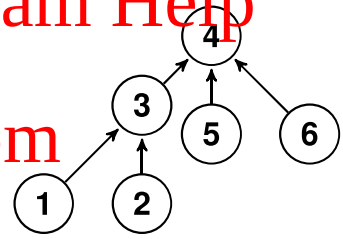
A. T_A



B. T_B



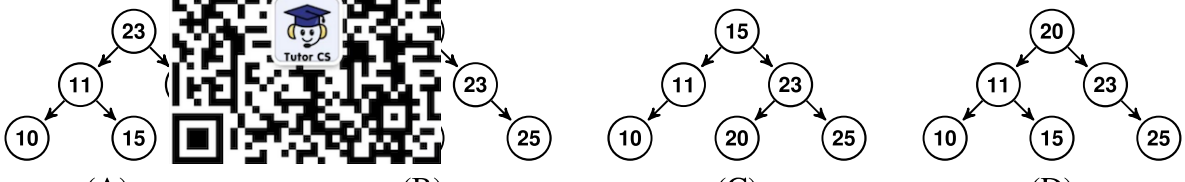
C. T_C



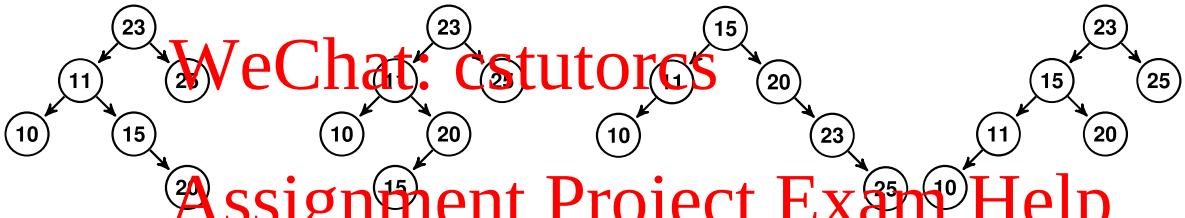
D. T_D

(e) (8 points) We want to insert a new key with value **20** in the AVL tree below.

Consider now th is the input tree from above).



(A) (B) (C) (D)



(E) (F) (G) (H)

Which of the following sequences of trees would represent the series of basic operations (i.e. BST insertion, rotation left and rotation right) as they would be executed by the AVL insertion method seen in class?

A. $A \to E \to H \to C$

B. $A \to H \to E \to C$

C. $A \to H \to D \to F$

D. $A \to B \to G \to D$

E. $A \to B \to F \to E$

F. $A \to E \to D \to B$