1. 1. It is possible to construct a BST from an unsorted array in $O(n)$ time.

   a) True b) False

   False, then we could sort in $O(n)$ time using an in-order traversal of the BST

   2. It is possible to convert an unsorted array into a minheap in $O(n)$ time.

   a) True b) False

   True, see CLRS p.153 BUILD-MAX-HEAP

   3. Aliens come from space and give you an algorithm to solve 3-SAT in $O(n^2)$ time. This means all problems in NP can be solved in $O(n^2)$ time.

   a) True b) Unknown

   Unknown, some problems in NP may not be reducible to 3-SAT in O(1) time

   4. What is the maximum possible height for a binary search tree with n nodes?

   a) $\log n$ b) $n-1$ c) $n/2$ d) $\sqrt{n}$

   B

   5. The problem of finding the longest simple path (*i.e.,* one with no repeated vertices) between vertices s and t in a weighted, directed graph exhibits optimal substructure.

   a) True b) False

   False, if s,u,v,w,x,t is the longest path from s to t, it does not mean x,t is the longest path from x to t.

   6. Some problems in NP can be solved in polynomial time

   a) True b) False

   True, P is a subset of NP

   7. The solution to the recurrence $T(n) = 2T(n/4) + \Theta(n^{0.3} \log n)$ is:

   a) $\Theta(\sqrt{n})$ b) $\Theta(\sqrt{n} \log n)$ c) $\Theta(n^{0.3} \log n)$ d) $\Theta(n^2)$

   A, case 1 of the master theorem since $\log_4 2 = 1/2 > 0.3$

   8. If $A \leq_P B$ and $B \leq_P C$ and there is a polynomial time algorithm for B, then (choose the *best* answer):

   a) $B \in P$ and $C \in P$ b) $B \in P$ c) $A \in P$ and $B \in P$ d) $A \in P$ and $B \in P$ and $C \in P$

   c, nothing can be said of problem C

   9. The array [21 18 15 16 17 1 2 8 14] represents a binary max-heap.

   a) True b) False

   True

   10. If $G = (V, E)$ is a tree, then:

   a) G has a cycle b) $|V| = |E| + 1$ c) G is disconnected d) $|V| = |E| - 1$

   B, Theorem B.2 on p1174 of CLRS

2. (a) Two keys collide with probability $\frac{1}{m}$. There are $\binom{n}{2} = O(n^2)$ pairs of elements, so the expected number of collisions is $O(\frac{n^2}{m})$. Choosing $m = O(n^2)$ makes it $O(1)$.

(b) Assume towards a contradiction that G has a cycle $v_1, v_2, ..., v_k, v_1$ and a topological ordering. Then, in the topological ordering, $v_1 < v_2 < ... < v_k < v_1$, implying $v_1 < v_1$. In other words, v_1 must come before itself in the topological ordering. This is a contradiction, implying that the graph does not have a topological order.

(c) $\delta(x_1) = 0$
$\delta(x_2) = -2$
$\delta(x_3) = -6$
$\delta(x_4) = -10$

3. (a) Sort the contracts from largest to smallest penalty in $O(n \log n)$ time. Then, build the vehicles in that order.

(b) Let $J = j_1, j_2, ..., j_n$ be an optimal solution, and let $G = g_1, g_2, ..., g_n$ be a solution from the greedy algorithm. If $j_1 = g_1$, we're done, so assume that $j_1 \neq g_1$. This implies that $p_{j_1} \leq p_{g_1}$. Additionally, since all vehicles are built under both G and J, there exists some $j_m = g_1$, where $m > 1$ and $p_{j_m} \geq p_{j_1}$. The total penalty amount for J is:

$$P(J) = \sum_{i=1}^{n} i \cdot p_{j_i} = 1p_{j_1} + 2p_{j_2} + ... + np_{j_n}$$

Since $p_{j_m} \geq p_{j_1}$, if we construct J' by swapping j_1 and j_m:

$$P(J') = P(J) - 1j_1 - mj_m + 1j_m + mj_1 \leq P(J)$$

Therefore, J' is an optimal solution that agrees with our greedy choice.

(c) After scheduling one vehicle for day one, we now have to schedule $n-1$ vehicles in $n-1$ days. This is the same problem, but smaller.

Let $J = j_1, j_2, ..., j_n$ be an optimal solution in which $j_1 = g_1$. Let $J' = j_2, ..., j_n$ be a solution to the smaller subproblem. Assume towards a contradiction that J' is not optimal. Then we can construct a better solution than J by replacing J' with an optimal solution. This is a contradiction, since J was chosen to be optimal.

4. Call Dijkstra's algorithm $|V|$ times, using each vertex in V as the start vertex. Store the results in a matrix $dist$, where $dist(u, v)$ is the shortest path length from u to v. This requires $O(|V||E| \log |V|)$ time if Dijkstra is implemented with a minheap-based priority queue.

For each pair of vertices (u, v), $dist(u, v)$ is the shortest path from u to v and $dist(v, u)$ is the shortest path from v to u. Therefore, $dist(u, v) + dist(v, u)$ is the length of the shortest cycle containing u and v. For each pair of vertices, compute this value and take the minimum. This is the shortest cycle in G. This takes $O(|V|^2)$ time.

Also note that $dist(u, u) = 0$. It is not the length of the shortest cycle containing u.

5. (a) Certificate: Boolean assignments to the n variables of Φ.

Verify: Walk the formula, evaluating each clause under the truth assignments in the certificate. Count how many evaluate to true and how many evaluate to false. Since the formula is walked once, the runtime is clearly $O(mn)$, since there are m clauses and a clause cannot contain more than one copy of the same variable.

(b) Given a 3-CNF formula Φ, we will construct another 3-CNF formula Φ' where Φ is satisfiable iff it is possible to satisfy exactly half of the clauses of Φ'. The formula Φ' has the same n variables as Φ, in addition to three new variables p, q, r. Construct Φ' as follows:

1) Add m copies of the clause $(p \vee \neg p \vee q)$ (Note: these are always true)

2) Add $2m$ copies of the clause $(p \vee q \vee r)$ (Note: these are all the same, and therefore all true or all false)

3) Finally, add the original m clauses of Φ to Φ'.

We now need to show that Φ is satisfiable $\iff$ it is possible to satisfy exactly half of the clauses of Φ'.

$\Rightarrow$: Assume Φ is satisfiable. Then the corresponding m clauses of Φ' are satisfiable. Additionally, the m clauses that are $(p \vee \neg p \vee q)$ are always satisfied. Choose $r = p = q = 0$ to make the other $2m$ clauses false, and exactly half of the clauses are satisfied.

$\Leftarrow$: Assume we have a truth assignment that makes exactly half of the clauses of Φ' true. No matter what this assignment is, the m clauses $(p \vee \neg p \vee q)$ are satisfied. Φ' has $4m$ clauses, so this means that exactly m additional clauses are satisfied. Since the $2m$ clauses $(p \vee q \vee r)$ would all be true together, making $3m$ clauses true, the m additional satisfied clauses must be the m clauses of Φ. This means the assignment satisfies Φ.