1 Part A (70 points)

1. Consider the problem of sorting an array $A[1,...,n]$ of integers. We presented an $O(n \log n)$-time algorithm in class and, also, proved a lower bound of $\Omega(n \log n)$ for any comparison-based algorithm.

(a) Give an efficient sorting algorithm for a boolean array $B[1,...,n]$. (3 Points)

(b) Give an efficient sorting algorithm for an array $C[1,...,n]$ whose elements are taken from the set $\{1,2,3,4,5\}$. (4 Points)

(c) Give an efficient sorting algorithm for an array $E[1,...,n]$ whose elements are distinct ($E[i] \neq E[j]$, for every $i \neq j \in \{1,...,n\}$); and are taken from the set $\{1,2,...,2n\}$. (10 Points)

(d) In case you designed linear-time sorting algorithms for (a-c) above, does it mean that the lower bound for sorting of $\Omega(n \log n)$ is wrong? Explain. (3 Points)

2. A Fibonacci number is a number that appears in the Fibonacci Sequence 1, 1, 2, 3, 5, 8, 13, .... The next number is found by adding the two numbers before it. Formally, $F_0 = 1$, $F_1 = 1$ and $F_{n+1} = F_n + F_{n-1}$, for $n \geq 2$. Define the set

$$FIBO = \{n \mid n \text{ is Fibonacci number}\}$$

Give an efficient algorithm for checking if a given number $n$ is a Fibonacci number (i.e., an algorithm that return 1, if $n \in FIBO$ and 0, otherwise). Analyze the correctness and the running time of your algorithm. (15 Points)

3. Graph Coloring. A vertex coloring is an assignment of colors (or labels) to each vertex of a graph such that no edge connects two identically colored vertices. More formally, a proper vertex coloring is an assignment $c : V \rightarrow S$ such that $c(v) \neq c(u)$, for every edge $(u,v) \in E$. The elements of $S$ are called colors; the vertices of one color form a color class. If $|S| = k$, we say that $c$ is a $k$-coloring (often we use $S = \{1,...,k\}$). A graph is $k$-colorable if it has a proper $k$-coloring. The chromatic number $\chi(G)$ is the smallest value of $k$ such that $G$ is $k$-colorable.

(a) Let $G$ be a graph where no node has degree larger than $\Delta$. Prove that $\chi(G) \leq \Delta + 1$. (Hint: design a $\Delta + 1$ coloring algorithm and prove its correctness.) (5 Points)

---

\[^1\]In a boolean array $B[1,...,n]$, each element $B[i]$ (for $i = 1,...,n$) is either 0; or 1.
Planar graphs are graphs that can be drawn on the plane such that no edges cross each other. Planar graphs are known to satisfy certain properties. For example, the number of edges in a connected planar graph will be proportional to the number of nodes. In particular, it is known that in a planar graph there are at most $3n$ edges ($|E| < 3n$) and hence $\sum_{v \in V} \deg(v) < 6n$.

- Give an efficient 6-coloring algorithm for planar graphs. Prove its correctness and analyze its running time. (10 Points)

(c) Give an efficient algorithm that takes as input a graph $G(V,E)$ and returns a proper 2-coloring assignment if $G$ is 2-colorable; otherwise returning the message “$G$ is not 2-colorable”. Prove the correctness and analyze the running time of your algorithm. (5 Points)

4. Prove that in any tree, there exists a node which if removed breaks the tree into connected components such that no connected component contains more than half the original nodes. (5 Points)

5. For a given graph $G(V,E)$. The distance $d_G(v,u)$ between $v$ and $u$ is the length in hops of the shortest path between $v$ and $u$. The diameter $D_G$ of a graph $G$ is the maximum distance among all pairs (of nodes) in $V$, i.e., $D_G = \max\{d_G(u,v) \mid v,u \in V\}$.

Let $a,b \in V$ be two nodes such that $d_G(a,b) = D_G$. Decide whether each of the following statements are correct and give a proof for each part.

(a) For every node $r \in V$ either $a$ or $b$ is a leaf in a BFS tree of $G$ rooted at $r$. (4 Points)

(b) Node $a$ is a leaf in any BFS tree of $G$ rooted at $b$. (3 Points)

(c) The depth of every BFS tree of $G$ is at least $D_G/2$. (The depth of a tree is the depth of its deepest leaf) (3 Points)
2 Part B (programming, 30 points)

Jim Bone, the Husky Dog secret agent, has finished his secret mission in the Kat Kingdom capital, and now has to return to the Dog District. However, the Kitties Guarding Boarders know that Jim Bone is in their country, and are attempting to stop him by spreading his picture to the kitty police at their train stations.

You must help Jim Bone escape the kitty capital by determining which trains he should take to leave the country. However, because the police are actively looking for him, Jim has decided that he cannot sit still at any one train station for too long. Thus you need to determine which trains he should take so that his layover is never too long.

**Input Format:**

Line 1 : \( N \ M \ T \)

- \( N \) is the max label of any train station (1 is the starting point, \( N \) is the destination outside the country). Jim Bone always starts at station 1 at time 0.
- \( M \) is the number of train rides which appear on the schedule.
- \( T \) is the max time Jim can spend at any one station without being caught by the Kitties Guarding Borders. Note that he is allowed to spend exactly \( T \) time steps at a station, but not a moment more.

Next \( M \) Lines: \( x \ y \ t_1 \ t_2 \).

- \( x \) is the station label where a train will be leaving from
- \( y \) is the station label where this train will arrive
- \( t_1 \) is the time the train will leave station \( x \).
- \( t_2 \) is the time the train will arrive at station \( y \).

**Output Format:**

"NO" if, no matter what trains Jim takes, he will not be able to reach station \( N \) without being caught.

"YES \( T_{\text{min}} \)" if he can escape, where \( T_{\text{min}} \) is the time that he arrives at station \( N \). If there are multiple such paths, you should find the one with smallest possible \( T_{\text{min}} \).

**Details of traveling:** At the very start, Jim is at station 1, at time 0. Suppose that Jim arrives at station \( x \) at time \( t \). Then he can move to station \( y \) if and only if:

- There is a train going from \( x \) to \( y \) at time \( t_1 \), where \( t_1 > t \), but \( t_1 - t \leq T \), the max time he can spend at the station.
- If there is such a train ride, labeled as \( x, y, t_1, t_2 \), then after taking this train, he will be at station \( y \) at time \( t_2 \). Furthermore, he will have spent \( t_1 - t \) time units waiting at station \( x \).