| | |
|---|---|
| COL351: Analysis and Design of Algorithms | Announced on: Aug 10, 2026 (Monday) |

# Tutorial Sheet 4

**Problems marked with ($\star$) will not be asked in the tutorial quiz.**

1. Let $G = (V, E)$ be an unweighted and undirected connected graph, and let BFS be run from a source vertex $s$. Let $\text{dist}[v]$ denote the minimum number of edges on any path from $s$ to $v$.

   A *BFS tree* is any spanning tree of the reachable vertices in which, for every reachable vertex $v$, the distance from $s$ to $v$ measured in the tree is equal to $\text{dist}[v]$.

   Identify which of the following statements are `TRUE`. Select all that apply and provide justification for why the selected options are `TRUE` and why the unselected options are `FALSE`.

   a) Every edge $(u, v) \in E$ satisfies $|\text{dist}[u] - \text{dist}[v]| \leqslant 1$ after BFS from $s$.

   b) Consider the edges $E' = \{(u, v) \subseteq E \mid \text{dist}[v] - \text{dist}[u] = 1\}$. Suppose we construct the edges $E'_{dir}$ by directing the edges in $E'$ from the vertex with lower $\text{dist}()$ value to the vertex with higher $\text{dist}()$ value. Then, the graph $G'_{dir} = (V, E'_{dir})$ contains at least one directed cycle.

   c) The number of distinct valid BFS trees (over all possible tie-breakings in BFS) is always polynomial in $|E|$.

   d) The number of distinct valid BFS trees (over all possible tie-breakings in BFS) is not always polynomial in $|V|$.

2. Let $G$ be a directed acyclic graph (DAG). Let $G^{\text{rev}}$ denote the graph obtained by reversing every edge of $G$.

   Identify which of the following statements are `TRUE`. Select all that apply and provide justification for why the selected options are `TRUE` and why the unselected options are `FALSE`.

   a) If $G$ has a unique topological ordering $T = (v_1, v_2, \ldots, v_n)$, then for every pair of consecutive vertices $(v_i, v_{i+1})$ in $T$, there must be a directed edge $v_i \to v_{i+1}$.

   b) If $T$ is a valid topological ordering of $G$, then the reversed sequence $T^{\text{rev}}$ is a valid topological ordering of the reversed graph $G^{\text{rev}}$.

   c) If $G$ has exactly $k$ distinct topological orderings, then $G^{\text{rev}}$ also has exactly $k$ distinct topological orderings.

   d) The number of distinct topological orderings of a DAG with $n$ vertices is always polynomial in $n$.

3. The game of Snakes and Ladders has a board with $n$ cells where you seek to travel from cell 1 to cell $n$. To move, a player throws a six-sided dice to determine how many cells forward they move. The board also contains snakes and ladders that connect certain pairs of cells. A player who lands on the mouth of a snake immediately falls back down to the cell at the other end. A player who lands on the base of a ladder immediately travels up to the cell at the top of the ladder. Suppose you have rigged the dice to give you full control of the number for each roll. Design an efficient algorithm to find the minimum number of dice throws to win.

4. Let $u$ and $v$ be two vertices in a directed graph $G = (V, E)$. Design a linear-time algorithm to find the number of distinct shortest paths (not necessarily vertex disjoint) between $u$ and $v$.

5. A *vertex cover* of an undirected graph $G = (V, E)$ is a subset of vertices $U \subseteq V$ such that each edge in $E$ is incident to at least one vertex of $U$. Design an efficient algorithm (using BFS or DFS) to find a minimum-size vertex cover if $G$ is a tree.

6. An *independent set* of an undirected graph $G = (V, E)$ is a subset of vertices $U \subseteq V$ such that no edge in $E$ is incident to two vertices of $U$. Design an efficient algorithm (using BFS or DFS) to find a maximum-size independent set if $G$ is a tree.

7. Given an undirected graph $G$, an *independent vertex cover* of $G$ is a subset of vertices that is both an independent set and a vertex cover of $G$. Design an efficient algorithm for testing whether $G$ contains an independent vertex cover.

8. ($\star$) In the 2SAT problem, you are given a set of clauses, each of which is the disjunction (logical "`OR`") of two literals. (A literal is a Boolean variable or the negation of a Boolean variable.) You would like to assign a value `TRUE` or `FALSE` to each of the variables so that all the clauses are satisfied, with at least one true literal in each clause. For example, if the input contains the three clauses $x_1 \vee x_2$, $\neg x_1 \vee x_3$, and $\neg x_2 \vee \neg x_3$, then one way to satisfy all of them is to set $x_1$ and $x_3$ to "`TRUE`" and $x_2$ to "`FALSE`".

   Design an algorithm that determines whether or not a given 2SAT instance has at least one satisfying assignment. (Your algorithm is responsible only for deciding whether or not a satisfying assignment exists; it need not exhibit such an assignment.) Your algorithm should run in $\mathcal{O}(m + n)$ time, where $m$ and $n$ are the number of clauses and variables, respectively.

9. ($\star$) Given a directed graph $G = (V, E)$ and any pair of vertices $u, v \in V$, let $\texttt{dist}(u, v)$ denote the length of the shortest directed path from $u$ to $v$ in $G$. If no such path exists, then define $\texttt{dist}(u, v) = +\infty$.

By means of an efficient algorithm, show that in any directed graph, there exists an *independent set* of vertices that can reach every other vertex in at most two steps. That is, design a polynomial-time algorithm that, given as input a directed graph $G = (V, E)$, computes a subset of vertices $S \subseteq V$ such that:

a) $\texttt{dist}(u, v) \geqslant 2$ whenever $u, v \in S$ and $u \neq v$, and

b) given any vertex $v \notin S$, there is a vertex $u \in S$ such that $\texttt{dist}(u, v) \leqslant 2$.

*Hint*: The independent set need not be of the maximum possible size. Can you think of an algorithm that lines up the vertices and computes the desired set by doing left-to-right and/or right-to-left passes?