

LaTeX Scribe Generation for Lecture Notes



Hello, please work as a scribe get
It is very important that you strict
Use the file "group_6_ITS_project
The final output should be valid
For now, provide only the LaTeX



Code snippet

```
\documentclass[11pt]{article}
\usepackage[utf8]{inputenc}
\usepackage[margin=1in]{geometry}
\usepackage{amsmath}
\usepackage{amssymb}
\usepackage{graphicx}
\usepackage{enumitem}
\usepackage{hyperref}

\titl{CSE400: Fundamentals of Probability in Computing \\ Lecture 10: Randomized Min-Cut Algorithm}
\author{Dhaval Patel, PhD}
\date{February 5, 2026}

\begin{document}

\maketitle

\section{Introduction to the Min-Cut Problem}
[cite_start]The minimum cut (min-cut) algorithm is utilized in various applications to solve problems related to network flow and connectivity. It helps in identifying the smallest set of edges whose removal disconnects a graph into two or more components. This concept is fundamental in fields such as computer networking, VLSI design, and operations research.

\subsection{Applications}
\begin{itemize}
    [cite_start]\item \textbf{Network Design:} Helps improve communication efficiency and optimize network infrastructure.
    [cite_start]\item \textbf{Communication Networks:} Useful for understanding network vulnerability to attacks and for designing secure communication protocols.
    [cite_start]\item \textbf{VLSI Design:} Used for partitioning circuits into smaller components to reduce manufacturing costs and power consumption.
\end{itemize}

\subsection{Definitions}
\begin{itemize}
    [cite_start]\item \textbf{Cut-set:} A set of edges whose removal breaks a graph into two or more connected components.
    [cite_start]\item \textbf{Min-Cut Problem:} Given a graph  $G=(V,E)$  with  $n$  vertices, the goal is to find a cut-set that separates the source vertex from the sink vertex while minimizing the number of edges removed.
    [cite_start]\item \textbf{Edge Contraction:} The primary operation in these algorithms. It removes an edge  $(u,v)$  and merges the vertices  $u$  and  $v$  into a single vertex.
\end{itemize}

\section{Max-Flow Min-Cut Theorem}
[cite_start]The theorem states: "In a flow network, the maximum amount of flow passing from the source to the sink is equal to the capacity of the minimum cut." This theorem provides a theoretical foundation for the development of efficient algorithms for solving min-cut problems.

\begin{itemize}
    [cite_start]\item \textbf{Capacity of a cut:} The sum of capacities of edges oriented from a vertex to another vertex in the cut-set.
    [cite_start]\item \textbf{Max Flow:} The largest possible flow from source  $S$  to sink  $T$ .
\end{itemize}

\section{Deterministic vs. Randomized Algorithms}

\subsection{Deterministic: Stoer-Wagner Algorithm}
[cite_start]This approach always guarantees an exact minimum cut[cite: 404].
\begin{itemize}
    [cite_start]\item \textbf{Logic:} A minimum cut is either the minimum  $s - t$ -cut of  $G$ , or the minimum  $s - t$ -cut of  $G'$ , where  $G'$  is obtained by removing all edges from  $G$  that have positive flow in the current solution.
\end{itemize}
```

```

[cite_start]\item \textbf{Complexity:} $O(VE + V^2 \log V)$ [cite: 406].  

\end{itemize}

\subsection{Randomized: Karger's Algorithm}
[cite_start]Randomized algorithms provide a probabilistic guarantee of success and may provide accurate  

\begin{itemize}
    [cite_start]\item \textbf{Efficiency:} Karger's algorithm has a time complexity of $O(V^2)$ [cite: 41]  

    [cite_start]\item \textbf{Probability of Success:} The algorithm outputs a min-cut set with probability  

    \item \textbf{Sensitivity:} It can be sensitive to the initial choice of edges. [cite_start]If critical  

\end{itemize}

\newpage

\section{Pseudocode}

\subsection{Deterministic Minimum Cut (Stoer-Wagner)}
\textbf{Algorithm 1: MinimumCutPhase(G, a)}  

\begin{enumerate}
    \item $A \leftarrow \{a\}$  

    \item \textbf{while} $A \neq V$ \textbf{do}  

        \item \textbf{return} the cut weight as the "cut of the phase".  

\end{enumerate}

\textbf{Algorithm 2: MinimumCut(G)}  

\begin{enumerate}
    \item \textbf{while} $|V| \geq 1$ \textbf{do}  

        \item Choose any $a$ from $V$.  

        \item Run \textit{MinimumCutPhase}(G, a).  

        \item \textbf{if} cut-of-the-phase < current minimum cut \textbf{then} store it as current minimum cut.  

        \item Shrink $G$ by merging the two vertices added last.  

        \item \textbf{return} the minimum cut.  

\end{enumerate}

\subsection{Recursive Randomized Min-Cut (Karger)}
\textbf{Algorithm 3: RECURSIVE-RANDOMIZED-MIN-CUT(G, $\alpha$)}  

\begin{itemize}
    \item \textbf{Input:} Undirected multigraph $G$ with $n$ vertices, integer constant $\alpha > 0$.  

    \item \textbf{if} $n \leq \alpha^3$ \textbf{then} return min-cut via brute force.  

    \item \textbf{else}  

        \item \textbf{for} $i \leftarrow 1$ \textbf{to} $a$ \textbf{do}  

            \item $G' \leftarrow$ multigraph obtained by applying $n - \lceil \frac{n}{\sqrt{\alpha}} \rceil$ contractions.  

            \item $C' \leftarrow$ \textit{RECURSIVE-RANDOMIZED-MIN-CUT}($G'$, $\alpha$).  

            \item \textbf{if} $i=1$ or $|C'| < |C|$ \textbf{then} $C \leftarrow C'$.  

        \item \textbf{return} $C$.  

\end{itemize}

\section{Comparison Summary}
\begin{table}[h]
\centering
\begin{tabular}{l|l|l|l}
\hline
Feature & Deterministic (Stoer-Wagner) & Randomized (Karger's) & \\
\hline
[cite_start]\textbf{Guarantee:} & Exact minimum cut [cite: 404] & Approximate with high probability [cite: 406] & \\
[cite_start]\textbf{Complexity:} & $O(VE + V^2 \log V)$ [cite: 406] & $O(V^2)$ [cite: 418] & \\
[cite_start]\textbf{Efficiency:} & Lower on large graphs [cite: 405] & Higher/Parallelizable & \\
\end{tabular}
\end{table}

\end{document}

```

Would you like me to generate a detailed summary of the edge contraction process or explain the Max-Flow Min-Cut theorem?