

# LaTeX Scribe Generation for Lecture Notes

Hi, you should behave as a professional scribe generator. Create a scribe as a LaTeX code of this file "L10\_Scribe" that you should not change the content in the file. Keep the content accurate. Follow the lecture's structure (1).pdf" as the reference for the intended scribe's format. Make it possible so that it can be directly converted



Certainly! Here is the LaTeX code for your lecture scribe, based on the content of the "L10\_S1" file and formatted according to your requirements:

Code snippet

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

\begin{document}

% --- Header Section ---
\begin{center}
    \textbf{\Large School of Engineering and Applied Science (SEAS), Ahmedabad University} \\
    \textbf{CSE 400: Fundamentals of Probability in Computing} \\
    \textbf{Lecture 10: Randomized Min-Cut Algorithm} \\
    \rule{\linewidth}{0.5pt}
\end{center}

\noindent \textbf{Lecturer:} Dhaval Patel, PhD \hfill \textbf{Date:} February 5, 2026 \\
\textbf{Topic:} Randomized Min-Cut Algorithm \hfill \textbf{Scribe:} Lecture Summary

\vspace{1em}

% --- Section 1: Outline ---
\section{Outline}
This lecture covers the fundamental concepts of the Min-Cut problem, comparing deterministic and random.
\begin{itemize}[noitemsep]
    \item The Min-Cut Problem: Definition and Applications.
    \item Max-Flow Min-Cut Theorem.
    \item Deterministic Min-Cut (Stoer-Wagner Algorithm).
    \item Randomized Min-Cut (Karger's Algorithm).
    \item Performance Comparison and Success Probabilities.
\end{itemize}

\section{The Min-Cut Problem}

\subsection{Why use Min-Cut?}
Min-cut algorithms are utilized in various domains to solve problems related to network connectivity, re
\begin{itemize}
    \item \textbf{Network Design:} Improves communication efficiency and optimizes network flow by find
    \item \textbf{Communication Networks:} Helps understand network vulnerability to failures and aids
    \item \textbf{VLSI Design:} Useful for partitioning circuits into smaller components to reduce inte
\end{itemize}

\subsection{What is a Min-Cut?}
\begin{itemize}
    \item \textbf{Cut-set:} A set of edges whose removal breaks a graph into two or more connected comp
```

\item \textbf{Min-cut Problem:} Given a graph  $G = (V, E)$  with  $n$  vertices, find a minimum cardinality cut.

\end{itemize}

\subsection{Edge Contraction}

The primary operation in randomized min-cut algorithms is \textbf{edge contraction}:

\begin{itemize}

\item It removes an edge  $(u, v)$  and merges vertices  $u$  and  $v$  into a single vertex.

\item All edges connecting  $u$  and  $v$  are eliminated.

\item All other edges are retained; the resulting graph may have parallel edges but no self-loops.

\end{itemize}

\section{Success and Failure in Min-Cut Runs}

\begin{itemize}

\item \textbf{Successful Run:} An iteration of the algorithm that correctly identifies the minimum cut.

\item \textbf{Unsuccessful Run:} An iteration where the algorithm fails to identify the minimum cut.

\end{itemize}

\section{Max-Flow Min-Cut Theorem}

The theorem states: "In a flow network, the maximum amount of flow passing from the source ( $s$ ) to the sink ( $t$ ) is equal to the minimum capacity of a cut separating  $s$  and  $t$ ."

\begin{itemize}

\item \textbf{Capacity of a Cut:} The sum of the capacity of edges oriented from a vertex in set  $X$  to a vertex in set  $Y$ .

\item \textbf{Minimum Cut:} The cut in the network with the smallest possible capacity.

\item \textbf{Maximum Flow:} The largest possible flow from source  $s$  to sink  $t$ .

\end{itemize}

\section{Deterministic Min-Cut Algorithm}

\subsection{Stoer-Wagner Algorithm}

This algorithm provides an exact solution. The core theorem states that for two vertices  $s$  and  $t$ , a minimum  $s$ - $t$  cut of  $G$  is equal to a minimum cut of  $G/\{s, t\}$  (the graph where  $s$  and  $t$  are merged).

\end{enumerate}

\paragraph{Complexity:}  $O(V \cdot E + V^2 \log V)$ .

\section{Randomized Min-Cut Algorithm}

\subsection{Why Randomized Algorithms?}

\begin{itemize}

\item Provide a probabilistic guarantee of success.

\item Often provide accurate estimates with fewer iterations.

\item Advantages: Efficiency, Parallelization, Approximation Guarantees, Robustness, and avoidance of worst-case scenarios.

\end{itemize}

\subsection{Karger's Randomized Algorithm}

Karger's algorithm uses random contraction steps to find a cut.

\paragraph{Theorem for Min-Cut Set:} The algorithm outputs a min-cut set with probability at least:  $P(\text{success}) \geq \frac{2}{n(n-1)}$

\paragraph{Complexity:}  $O(V^2)$ .

\section{Comparison: Deterministic vs. Randomized}

Feature	Deterministic (Stoer-Wagner)	Randomized (Karger's)
Guarantee	Always finds exact minimum cut	Approximate min-cut with high probability
Complexity	$O(V \cdot E + V^2 \log V)$	$O(V^2)$
Scalability	Higher complexity for large graphs	Generally more efficient for large graphs

\end{table}

\section{Python Simulation}

A simulation of the Randomized Min-Cut algorithm can be found in the lecture 10 Campuswire post (File: RandomizedMinCut.py).

\vspace{2em}

\begin{center}

\textbf{End of Scribe}

\end{center}

