Comparative Analysis of Priority Queues in Graph Algorithms

First Last
Department / University
City, State, Country
email@domain.com

***Abstract*—Abstract goes here.**

***Index Terms*—Dijkstra, Prim, pairing heap, Fibonacci heap, priority queue, experimental analysis**

I. Introduction

Blah Blah Blah.

II. Background

A. Dijkstra's Algorithm

B. Prim's Algorithm

C. Priority Queue Operations

D. Theoretical Complexity

Heap Type	Insert	Extract-Min	Decrease-Key
Binary Heap			
Pairing Heap			
Fibonacci Heap			

TABLE I
ASYMPTOTIC TIME COMPLEXITIES (FILL IN LATER).

III. Implementation

A. System Design

B. Graph Representation

C. Pairing Heap

D. Fibonacci (or Binomial) Heap

E. Instrumentation

IV. Experimental Design

A. Environment

B. Graph Types

- Random graphs (sparse vs dense)
- Grid graphs
- Synthetic worst-case graphs

C. Metrics Collected

Blah Blah Blah.

V. Results

A. Total Runtime

Fig. 1. Runtime comparison (placeholder).

B. Operation Counts

C. Time Breakdown: Extract-Min vs Decrease-Key

VI. Discussion

Answer:

- Do Fibonacci heaps provide practical benefits?
- How do pairing heaps compare in practice?
- Which algorithm benefits more (Dijkstra vs Prim)?
- How does graph structure affect performance?
- Why do theory and practice differ?

VII. Threats to Validity

Measurement noise, graph generation bias, implementation constant factors.

VIII. Conclusion

Summary + future work.