1 Mil gap log()

```
#4 args
import numpy as np

def parse_latencies(file_path):
    latencies = []
    with open(file_path, 'r') as f:
        for line in f:
            if line.startswith("Latency:"):
                try:
                    _, value = line.strip().split(":")
                    latencies.append(float(value.strip()))
                except ValueError:
                    pass  # skip malformed lines
    return latencies

def compute_percentiles(latencies, percentiles):
    return np.percentile(latencies, percentiles)

# Usage
file_path = 'latency_1mic.txt'  # Replace with your actual file name
latencies = parse_latencies(file_path)
percentiles = [25,50, 90, 95, 99]
results = compute_percentiles(latencies, percentiles)

for i, p in enumerate(percentiles):
    print("{}th percentile latency: {}".format(p, results[i]))
```

```
25th percentile latency: 41.0
50th percentile latency: 83.0
90th percentile latency: 125.0
95th percentile latency: 125.0
99th percentile latency: 166.0
```

```
#8 args
import numpy as np

def parse_latencies(file_path):
    latencies = []
    with open(file_path, 'r') as f:
        for line in f:
            if line.startswith("Latency:"):
                try:
                    _, value = line.strip().split(":")
                    latencies.append(float(value.strip()))
                except ValueError:
                    pass  # skip malformed lines
    return latencies

def compute_percentiles(latencies, percentiles):
    return np.percentile(latencies, percentiles)

# Usage
file_path = 'latency_8.txt'  # Replace with your actual file name
latencies = parse_latencies(file_path)
percentiles = [25,50, 90, 95, 99]
results = compute_percentiles(latencies, percentiles)

for i, p in enumerate(percentiles):
    print("{}th percentile latency: {}".format(p, results[i]))
```

```
25th percentile latency: 83.0
50th percentile latency: 83.0
90th percentile latency: 125.0
95th percentile latency: 125.0
99th percentile latency: 250.0
```

100 mic gap

```
#8 args
import numpy as np

def parse_latencies(file_path):
    latencies = []
    with open(file_path, 'r') as f:
        for line in f:
            if line.startswith("Latency:"):
                try:
                    _, value = line.strip().split(":")
                    latencies.append(float(value.strip()))
                except ValueError:
                    pass  # skip malformed lines
    return latencies

def compute_percentiles(latencies, percentiles):
    return np.percentile(latencies, percentiles)

# Usage
file_path = 'latency.txt'  # Replace with your actual file name
latencies = parse_latencies(file_path)
percentiles = [25,50, 90, 95, 99]
results = compute_percentiles(latencies, percentiles)

for i, p in enumerate(percentiles):
    print("{}th percentile latency: {}".format(p, results[i]))
```

```
25th percentile latency: 83.0
50th percentile latency: 83.0
90th percentile latency: 125.0
95th percentile latency: 125.0
99th percentile latency: 250.0
```