

---

# Priority Queue Comparison Report

---

HEAP VS. LIST

SAMY MASADI  
CSCI 423  
MARCH 22, 2019

# Contents

<b>1</b>	<b>Summary</b>	<b>1</b>
<b>2</b>	<b>Methodology</b>	<b>2</b>
2.1	Heap Pseudocode . . . . .	2
<b>3</b>	<b>Performance Reports</b>	<b>3</b>
3.1	Modules: Size 10 . . . . .	3
3.2	Modules: Size 50 . . . . .	3
3.3	Modules: Size 500 . . . . .	3
3.4	Modules: Size 1000 . . . . .	3
3.5	Modules: Size 5000 . . . . .	3
3.6	Modules: Size 10000 . . . . .	4
3.7	Modules: Size 50000 . . . . .	4
3.8	Job Scheduler Test Results . . . . .	4
<b>4</b>	<b>Complexity Analysis</b>	<b>5</b>
4.1	Time Efficiency . . . . .	5
<b>5</b>	<b>Schedule</b>	<b>6</b>
<b>6</b>	<b>Screenshot Showcase</b>	<b>7</b>

# 1 Summary

This program measures the performance of priority queues using a heap structure and using a list structure. It generates several reports of time performance for each individual module for lists of several sizes. It also creates a report for a overall job scheduler using heap and list.

From the reports, we can conclude that enqueueing and dequeing in a heap is  $O(\log n)$  because the task of heapifying an item is divide in two. Enqueueing a list is constant because it is unsorted and order does not matter. Dequeueing a list is  $O(n)$  because it must search for the highest priority before removing it. For the job schedulers, the heap has better performance with  $O(n \log n)$  because it must enqueue and dequeue for all items. The list is  $O(n^2)$  because it must search all items for every item in the queue.

## 2 Methodology

Note: Pseudocode for heap modules sourced for CSCI 423 lecture slides.

### 2.1 Heap Pseudocode

```
enQueue:
    append(item)
    Reheapify Sift up:
        If parent is greater:
            swap(parent, item)
```

```
deQueue:
    swap(first, last)
    pop(last)    Reheapify:
        If child is smaller:
            swap(child, item)
```

## 3 Performance Reports

### 3.1 Modules: Size 10

Module	Heap	List
_init_	0.000016	0.000002
enQueue	0.000013	0.000004
deQueue	0.000005	0.000004
sneakAPeek	0.000001	0.000002
isEmpty	0.000001	0.000001
size	0.000001	0.000001

### 3.2 Modules: Size 50

Module	Heap	List
_init_	0.000057	0.000002
enQueue	0.000009	0.000003
deQueue	0.000006	0.000008
sneakAPeek	0.000001	0.000007
isEmpty	0.000001	0.000001
size	0.000000	0.000000

### 3.3 Modules: Size 500

Module	Heap	List
_init_	0.000590	0.000003
enQueue	0.000009	0.000003
deQueue	0.000009	0.000058
sneakAPeek	0.000001	0.000058
isEmpty	0.000001	0.000001
size	0.000001	0.000000

### 3.4 Modules: Size 1000

Module	Heap	List
_init_	0.001177	0.000004
enQueue	0.000009	0.000004
deQueue	0.000011	0.000146
sneakAPeek	0.000001	0.000139
isEmpty	0.000001	0.000001
size	0.000000	0.000000

### 3.5 Modules: Size 5000

Module	Heap	List
_init_	0.005858	0.000008
enQueue	0.000009	0.000003
deQueue	0.000012	0.000658
sneakAPeek	0.000001	0.000657
isEmpty	0.000001	0.000001
size	0.000001	0.000001

### 3.6 Modules: Size 10000

Module	Heap	List
_init_	0.012278	0.000052
enQueue	0.000010	0.000033
deQueue	0.000014	0.001183
sneakAPeek	0.000001	0.001175
isEmpty	0.000001	0.000001
size	0.000000	0.000000

### 3.7 Modules: Size 50000

Module	Heap	List
_init_	0.058573	0.000193
enQueue	0.000010	0.000006
deQueue	0.000018	0.005876
sneakAPeek	0.000001	0.006060
isEmpty	0.000001	0.000001
size	0.000001	0.000001

### 3.8 Job Scheduler Test Results

Size	Heap	List
10	0.002792	0.000860
50	0.003096	0.001290
500	0.007834	0.031011
1000	0.013855	0.119394
5000	0.069740	2.958742
10000	0.144353	11.856617
50000	0.817438	297.10306

## 4 Complexity Analysis

### 4.1 Time Efficiency

Module	Heap	List
<code>_init_</code>	$O(n \log n)$	$O(n)$
<code>enqueue</code>	$O(\log n)$	$O(1)$
<code>dequeue</code>	$O(\log n)$	$O(n)$
<code>sneakAPeek</code>	$O(1)$	$O(n)$
<code>isEmpty</code>	$O(1)$	$O(1)$
<code>size</code>	$O(1)$	$O(1)$
<code>scheduler</code>	$O(n \log n)$	$O(n^2)$

## 5 Schedule

March 3: Begin work on PQ\_Heap: `_init_`; `enQueue`; `deQueue`

Completed: none

March 10: Finish PQ\_Heap; start PQ\_List: `_init_`; `enQueue`; `deQueue`

Completed: none

March 17: Finish PQ\_List; Conduct time trials; Begin Report

Completed: none

March 22: Complete Report

Completed: PQ\_Heap, PQ\_List, Conducted time trials

Report completed March 24



## 6 Screenshot Showcase

Heap vs. List Performance Report  
Number of items: 10

Heap initialized.  
Contents: 4389, 25218, 15718, 33560, 27288, 39204, 15907, 35520, 40405, 31641  
Size: 10

EnQueue item of priority 1500.  
Parent: n/a  
Children: 4389, 15718  
EnQueue item completed.  
DeQueue highest priority item completed.  
Peak at highest priority item completed.  
Check for empty queue completed.  
Check size completed.

List initialized.  
Contents: 31641, 33560, 39204, 15718, 25218, 15907, 4389, 35520, 40405, 27288  
Size: 10

EnQueue item of priority 1500.  
Parent: n/a  
Children: n/a  
EnQueue item completed.  
DeQueue highest priority item completed.  
Peek at highest priority item completed.  
Check for empty queue completed.  
Check size completed.

Performance Results Table (measurements in seconds)

Module	Heap	List
__init__	0.000016	0.000002
enQueue	0.000013	0.000004
deQueue	0.000005	0.000004
sneakAPeek	0.000001	0.000002
isEmpty	0.000001	0.000001
size	0.000001	0.000001

Figure 1: report10.txt: Output report for 10 items

## Heap vs. List Performance Report

Number of items: 50000

Heap initialized.

Contents: 2, 4, 5, 7, 7, 16, 6, 8, 17, 33...

Size: 50000

EnQueue item of priority 1500.

Parent: 1134

Children: 14950, 15732

EnQueue item completed.

DeQueue highest priority item completed.

Peak at highest priority item completed.

Check for empty queue completed.

Check size completed.

List initialized.

Contents: 34101, 40963, 48844, 30250, 33533, 47355, 7032, 21318, 4112, 23067...

Size: 50000

EnQueue item of priority 1500.

Parent: n/a

Children: n/a

EnQueue item completed.

DeQueue highest priority item completed.

Peek at highest priority item completed.

Check for empty queue completed.

Check size completed.

### Performance Results Table (measurements in seconds)

Module	Heap	List
__init__	0.058573	0.000193
enQueue	0.000010	0.000006
deQueue	0.000018	0.005876
sneakAPeek	0.000001	0.006060
isEmpty	0.000001	0.000001
size	0.000001	0.000001

Figure 2: report50000.txt: Output report for 50000 items

---

Job Scheduler Test Results  
Time measurements are in seconds.

Size	Heap	List
10	0.002792	0.000860
50	0.003096	0.001290
10	0.002792	0.000860
500	0.007834	0.031011
1000	0.013855	0.119394
5000	0.069740	2.958742
10000	0.144353	11.856617
50000	0.817438	297.10306

Figure 3: scheduler\_results.txt output file