1 Overview

At first we discuss the importance of certain elements in the algorithm which are required for the fastest output possible. Many of these elements require special subsets of certain data structures to in order to achieve an optimized run time. The absolute worst case scenario for the marriage stable problem is $\bigcirc(n^2)$, has we will see that by use of powerful data structures we can far surpass the worst case. In order to discuss the running time of the actual running time of the Java code, we must understand the common running times of the individual structures used in it.

2 Code(Java)

```
private void applyGs(){
Male m;
Female f;

long start,end;
double duration;
start =System.nanoTime();
while(singleMaleList.size()!=0){

m = singleMaleList.remove();
f = m.preferenceQueue.remove();
if(f.engagedTo==null){
f.engagedTo=m;
}
else{
if(f.prefernceList.indexOf(f.engagedTo) < f.prefernceList.indexOf(m)){
singleMaleList.add(m);
}
else{
singleMaleList.add(f.engagedTo);
f.engagedTo=m;
}
}
}
assembleFinalPairs();
}
```

```
private class Male {
private String name;
private Queue<Female> preferenceQueue;

....
}

private class Female {
private Male engagedTo;
private String name;
private LinkedList<Male> prefernceList;
...
}
```

3 Common run times

Running Times of the two main Data Structures used in the Code.

3.1 LinkedList

Method	Run Time
Indexing	$\bigcirc(n)$
insert/delete(not at the start or end)	$searchtime + \bigcirc(1)$

3.2 HashMap

Method	Run Time
Indexing	$\bigcirc(n)$
insert	$\bigcirc(1)$
remove(n=total elements k=current index)	$\bigcirc(1 + n/k)$

4 Trial Results

Ran the code for 11 different trials. The input files were at random, meaning they were not set for the worst case scenario where women and men have the exact opposite preference lists.

n (Size of the input)	Time in seconds
50	$7.834 * 10^-4$
100	$1 * 10^-3$
150	$2.59 * 10^-3$
200	$8.95 * 10^-3$
250	$9.39 * 10^-3$
500	$9.69 * 10^-3$
1000	$1.17 * 10^-2$
1500	$3.72 * 10^-2$
2000	$6.87 * 10^-2$
2500	$6.67 * 10^-2$
5000	$2.224 * 10^-1$

lack of increase in running time from n = 250 to 500, and from 2000 to 2500 is based on the input file being near best case scenario rather then the later.