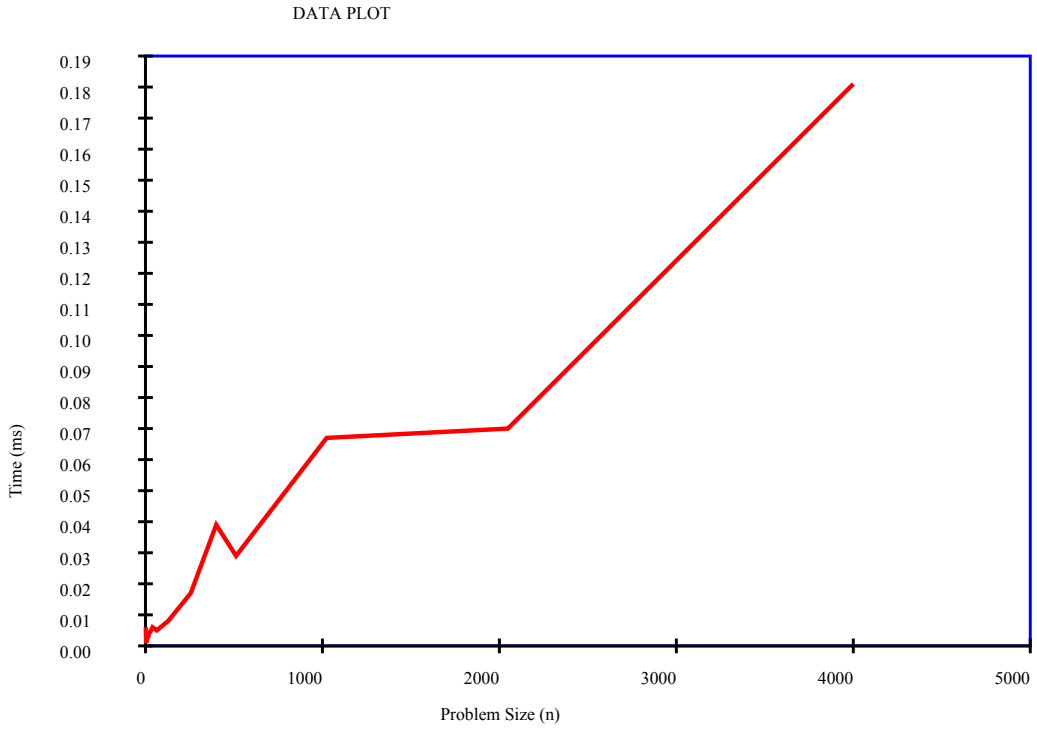
Report for Ruihong Zhang's Program 3

	TimeComplexity
min()	
removeMin()	
insert(K,V)	
remove(Entry)	
replaceKey(Entry,K)	
replaceValue(Entry,V)	

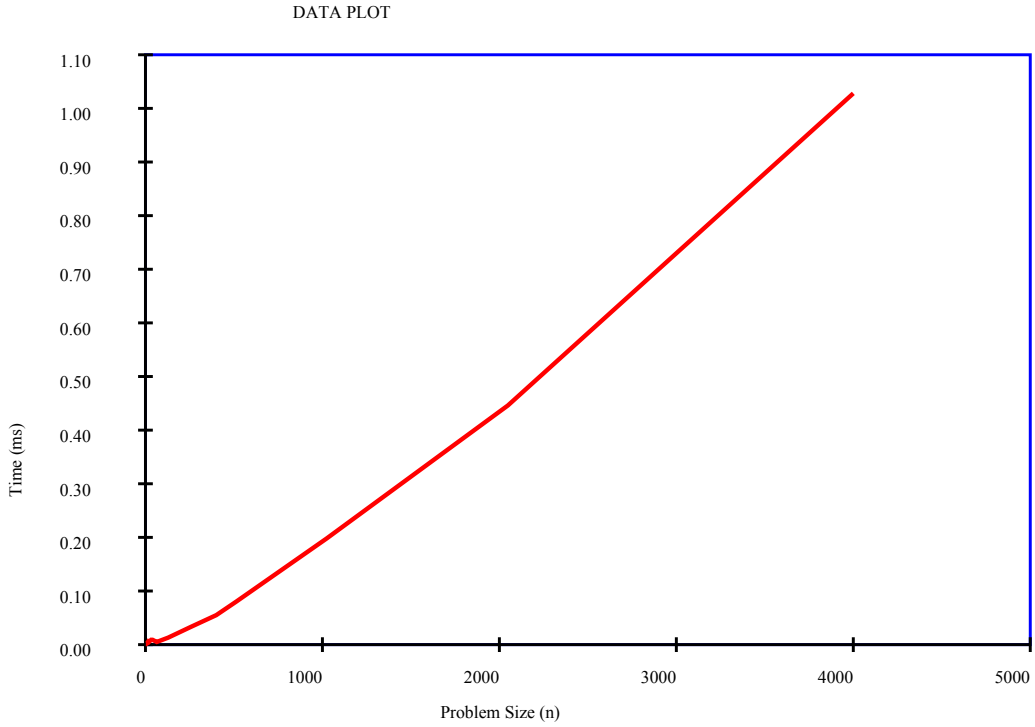
Heap Time Complexity Annotations (above table) _____ / 8.0

	TimeComplexityAmortized
insert(K,V)	

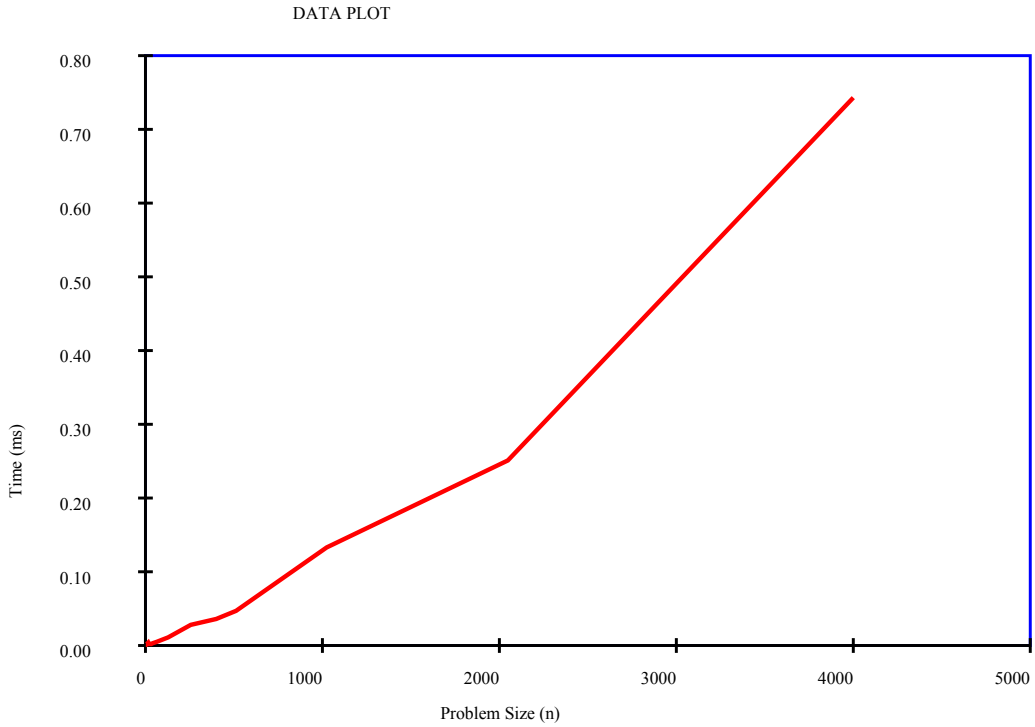
Heap Time Complexity Annotations (above table) _____ / 2.0



Timing test for Heap insert() (above graph) _____ / 5.0



Timing test for Heap remove() (above graph) _____ / 5.0



Timing test for Heap replaceKey() (above graph) _____ / 5.0

Heap tests 65.0 / 65.0

Empty PQ: 7.22 / 7.22

Verifying size()=0

Test Successful -0.00

Verifying isEmpty() is true

Test Successful -0.00

Verifying min() on Empty PQ returns null

Test Successful -0.00

Verifying removeMin() on Empty PQ returns null

Test Successful -0.00

One Item PQ - [50, "A"]: 7.22 / 7.22

Verifying size()=1

Test Successful -0.00

Verifying isEmpty() is false

Test Successful -0.00

Verifying min() returns an instance of Entry<K, V>

Test Successful -0.00

Verifying min().getKey() = 50

Test Successful -0.00

Verifying min().getValue() = "A"

Test Successful -0.00

Verifying removeMin() returns an instance of Entry<K, V>

Test Successful -0.00

Verifying removeMin().getKey() = 50

Test Successful -0.00

Verifying removeMin().getValue() = "A"

Test Successful -0.00

Verifying removeMin(); size() = 0

Test Successful -0.00

Verifying removeMin(); isEmpty() is true

Test Successful -0.00

Multiple Item PQ - [1,"A"], [2,"B"], [3,"C"], [4,"D"]: 7.22 / 7.22

Verifying 1st removeMin().getKey() = 1

Test Successful -0.00

Verifying 1st removeMin().getValue() = "A"

Test Successful -0.00

Verifying 2nd removeMin().getKey() = 2

Test Successful -0.00

Verifying 2nd removeMin.getValue() = "B"

Test Successful -0.00

Verifying 3rd removeMin().getKey() = 3

Test Successful -0.00

Verifying 3rd removeMin.getValue() = "C"

Test Successful -0.00

Verifying 4th removeMin().getKey() = 4

Test Successful -0.00

Verifying 4th removeMin.getValue() = "D"

Test Successful -0.00

Multiple Item PQ - [4,"D"], [3,"C"], [2,"B"], [1,"A"]: 7.22 / 7.22

Verifying 1st removeMin().getKey() = 1

Test Successful -0.00

Verifying 1st removeMin().getValue() = "A"

Test Successful -0.00

Verifying 2nd removeMin().getKey() = 2

Test Successful -0.00

Verifying 2nd removeMin.getValue() = "B"

Test Successful -0.00

Verifying 3rd removeMin().getKey() = 3

Test Successful -0.00

Verifying 3rd removeMin.getValue() = "C"

Test Successful -0.00

Verifying 4th removeMin().getKey() = 4

Test Successful -0.00

Verifying 4th removeMin.getValue() = "D"

Test Successful -0.00

Multiple Item PQ - [4,"D"], [2,"B"], [3,"C"], [1,"A"]: 7.22 / 7.22

Verifying 1st removeMin().getKey() = 1

Test Successful -0.00

Verifying 1st removeMin().getValue() = "A"

Test Successful -0.00

Verifying 2nd removeMin().getKey() = 2

Test Successful -0.00

Verifying 2nd removeMin.getValue() = "B"

Test Successful -0.00

Verifying 3rd removeMin().getKey() = 3

Test Successful -0.00

Verifying 3rd removeMin.getValue() = "C"

Test Successful -0.00

Verifying 4th removeMin().getKey() = 4

Test Successful -0.00

Verifying 4th removeMin.getValue() = "D"

Test Successful -0.00

Multiple Item Maximum PQ - [4,"D"], [2,"B"], [3,"C"], [1,"A"]: 7.22 / 7.22

Verifying 1st removeMin().getKey() = 4

Test Successful -0.00

Verifying 1st removeMin().getValue() = "D"

Test Successful -0.00

Verifying 2nd removeMin().getKey() = 3

Test Successful -0.00

Verifying 2nd removeMin.getValue() = "C"

Test Successful -0.00

Verifying 3rd removeMin().getKey() = 2

Test Successful -0.00

Verifying 3rd removeMin.getValue() = "B"

Test Successful -0.00

Verifying 4th removeMin().getKey() = 1

Test Successful -0.00

Verifying 4th removeMin.getValue() = "A"

Test Successful -0.00

Five Hundred PQ: 7.22 / 7.22

Insert [1,"A"], [2,"A"],...,[500,"A"]. Verifying removeMin() removes keys 1, 2,...,500

Test Successful -0.00

Insert [500,"A"], [499,"A"],...,[1,"A"]. Verifying removeMin() removes keys 1, 2,...,500

Test Successful -0.00

Very Large PQ: 7.22 / 7.22

Inserting 2048 items, where first item is 1234 and successive items are computed with: xn=(1021*x(n-1)+50)%2048

Test Successful -0.00

Inserting 4096 items, where first item is 1234 and successive items are computed with: xn=(1021*x(n-1)+50)%2048

Test Successful -0.00

Inserting 8192 items, where first item is 1234 and successive items are computed with: xn=(1021*x(n-1)+50)%2048

Test Successful -0.00

Adaptable PQ - [1,"A"], [2,"B"], [3,"C"], [4,"D"]: 7.22 / 7.22

Verifying replacing the entry <1,D> with value DD, the removeMin() will have getValue() DD

Test Successful -0.00

Verifying replacing the entry <3,C> with value CC, the third removeMin() will have getValue() CC

Test Successful -0.00

Verifying removing the entry <1,A>, min().getKey() = 2

Test Successful -0.00

Verifying removing the entry <1,A>, size() = 3

Test Successful -0.00

Verifying removing the entry <3,C>, min().getKey() = 1

Test Successful -0.00

Verifying removing the entry <3,C>, size() = 3

Test Successful -0.00

Verifying removing the entry <3,C>, then replace e4 with key 0, min() = 3

Test Successful -0.00

Verifying replacing the entry <1,A> to <10,A> min().getKey() = 2

Test Successful -0.00