delete_node.R

Preston

2020-05-23

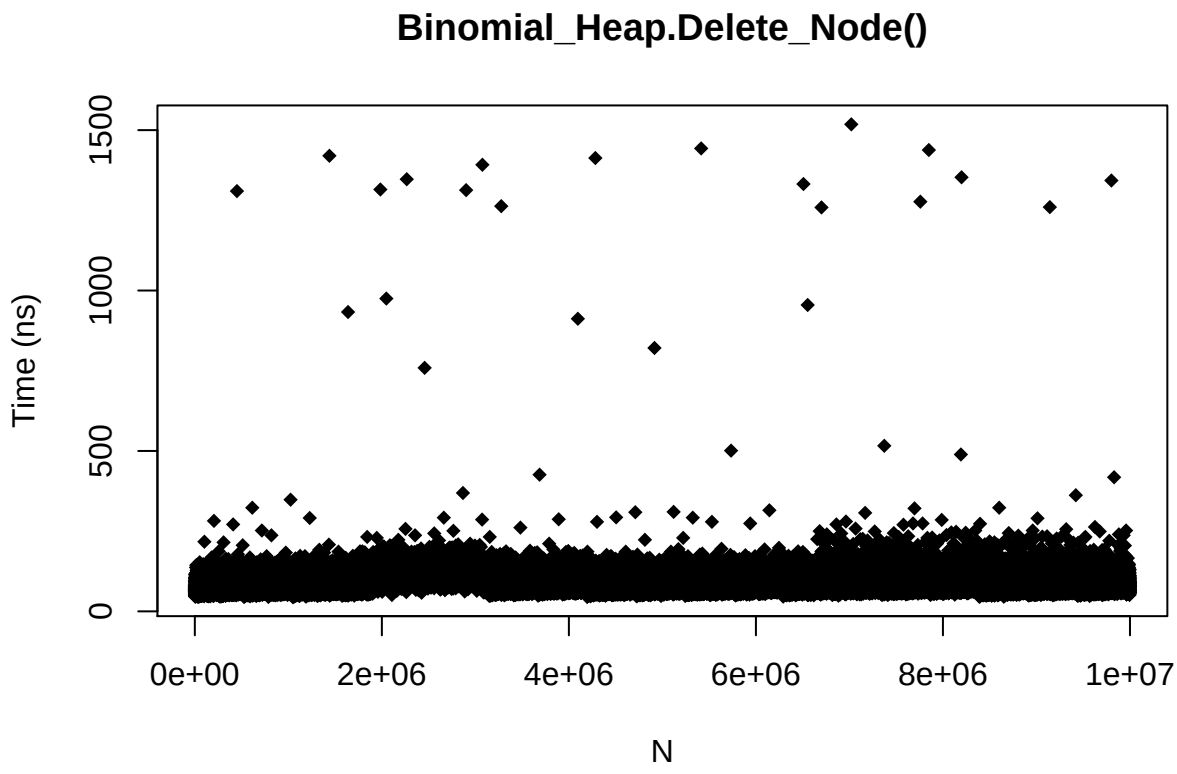
```
# Preston Dunton
# CS320 Honors Option
# May 23, 2020
# pdunton@rams.colostate.edu

# Delete_Node() in a binomial heap should be O(logn)

delete_node_binomial = read.csv("./delete_node_binomial.csv")
attach(delete_node_binomial)
```

```
## The following object is masked from package:base:
##
##     T
```

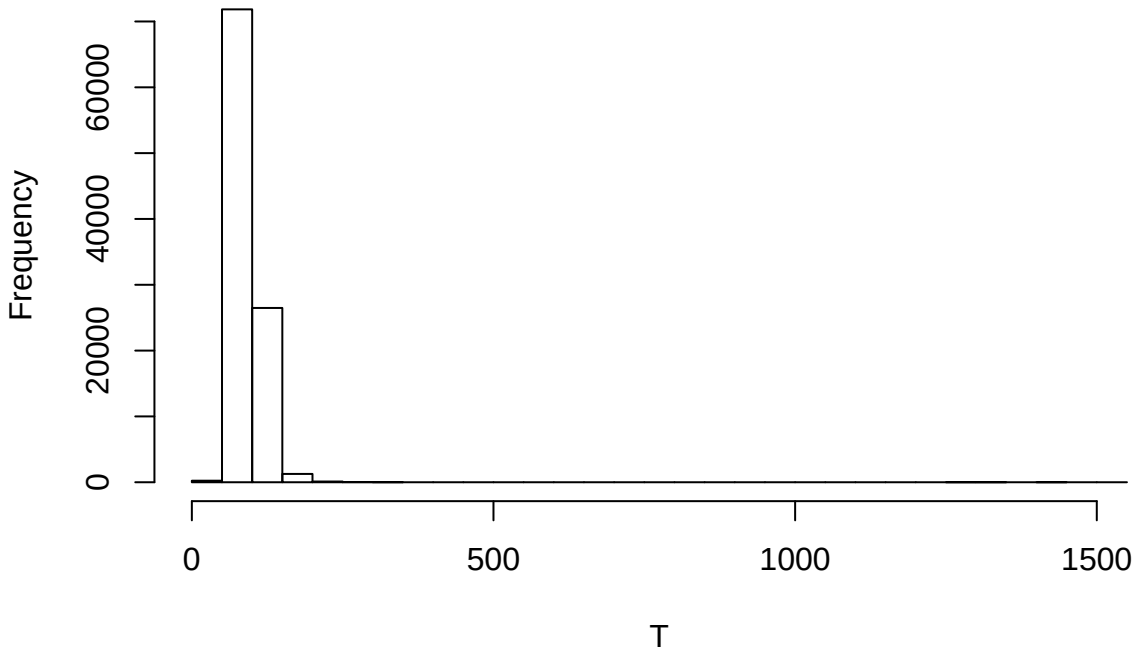
```
summary(T)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   44.00   69.00   83.00   88.43  103.00 1518.00
```

```
# min 44
# q1 69
# median 83
# mean 88.43
# q3 103
# max 1518
plot(N,T,pch=18,xlab="N",ylab="Time (ns)",main="Binomial_Heap.Delete_Node()")
```




```
hist(T,breaks=30)
```




```
# Let's see if we can remove some outliers

quantile(T,seq(0,1,0.1))
```

```
##   0%  10%  20%  30%  40%  50%  60%  70%  80%  90% 100%
##   44   61   67   72   77   83   91   99  108  121 1518
```

```
quantile(T,seq(0.9,1,0.01))
```

```
##  90%  91%  92%  93%  94%  95%  96%  97%  98%  99% 100%
##  121  123  125  127  129  132  135  139  146  156 1518
```

```
quantile(T,seq(0.99,1,0.001))
```

```
##      99%    99.1%    99.2%    99.3%    99.4%    99.5%    99.6%    99.7%
##  156.000  158.000  161.000  163.000  167.000  172.000  177.000  184.000
##    99.8%    99.9%     100%
##  196.000  227.002 1518.000
```

```
# Let's separate the top 0.1% and analyze

# Top 0.1%
sum(T>227) # There are 100 outliers
```

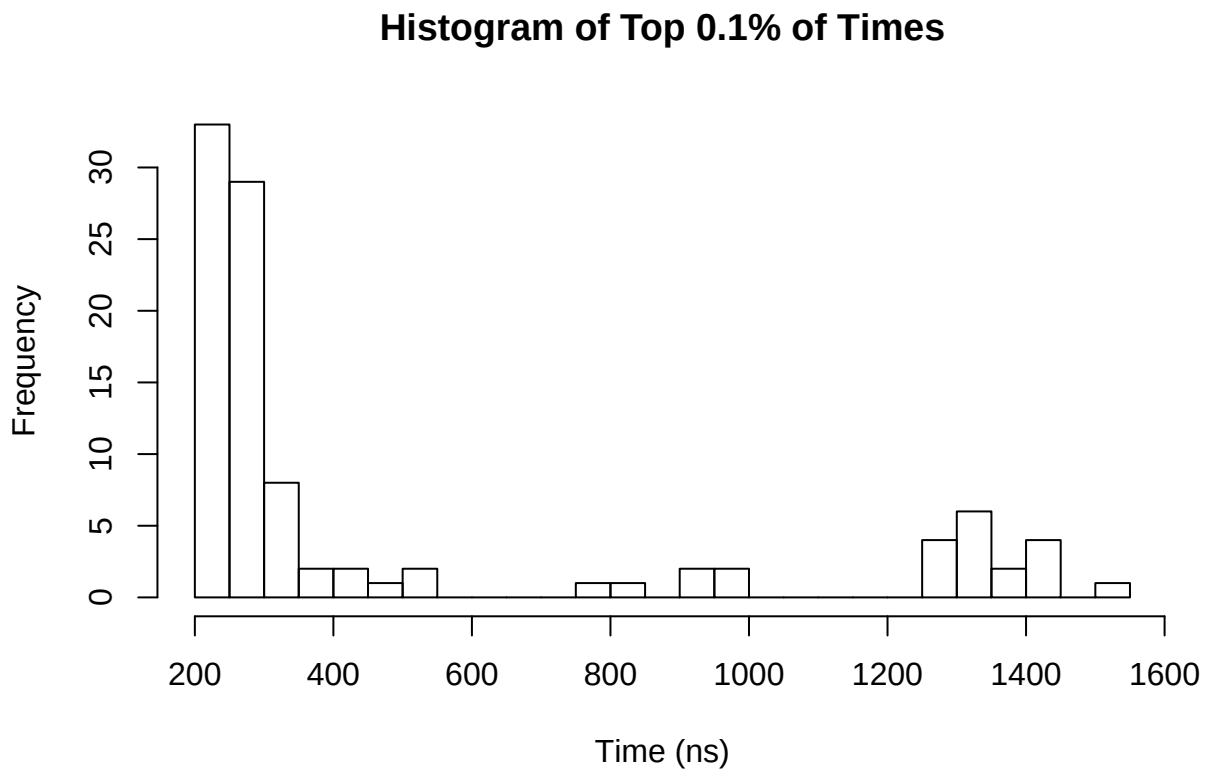
```
## [1] 100
```

```
summary(T[which(T>227)])
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
```

```
##   228.0   243.8   276.5   497.1   492.0  1518.0
```

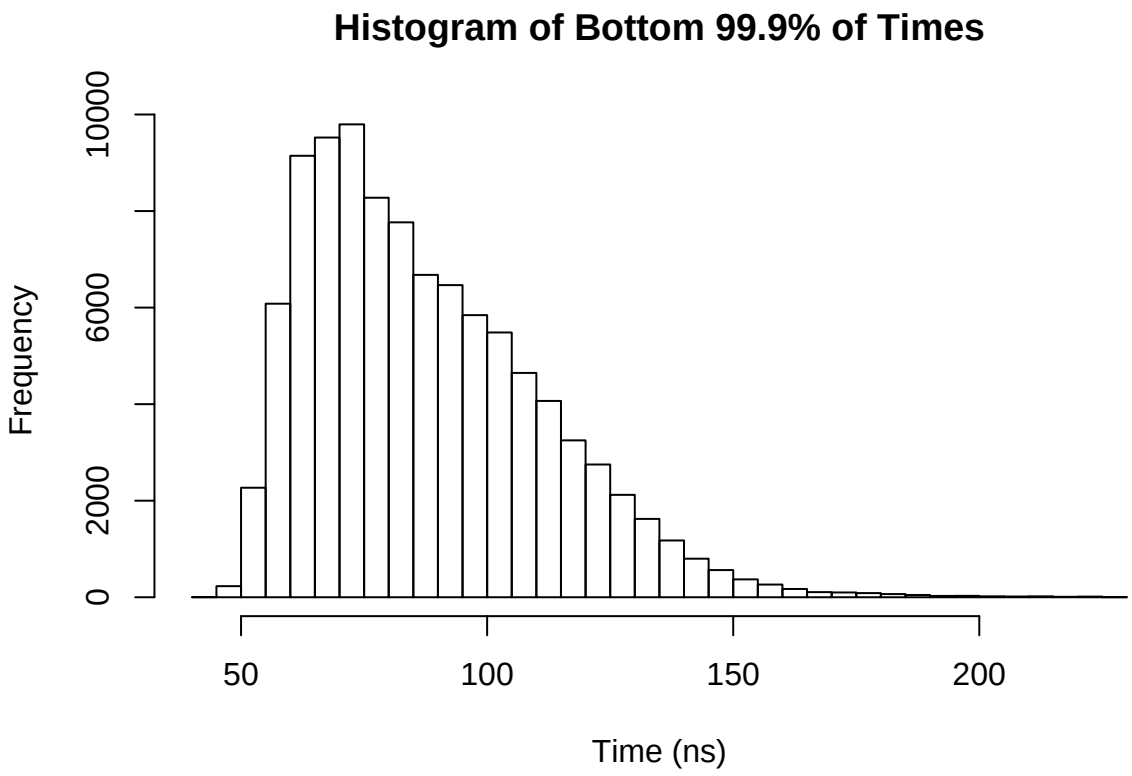
```
# min 228
# q1 243
# median 276.5
# mean 497.1
# q3 492
# max 1518
hist(T[which(T>227)],main="Histogram of Top 0.1% of Times",xlab="Time (ns)",breaks=30)
```




```
# Bottom 99.9%
summary(T[which(T<=227)])
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   44.00   69.00   83.00   88.02  103.00  227.00
```

```
# min 44
# q1 69
# median 83
# mean 88.02
# q3 103
# max 227
hist(T[which(T<=227)],main="Histogram of Bottom 99.9% of Times",xlab="Time (ns)",breaks=30)
```




```
# 99.9% of deletions take less than 227 ns
# There also appears to be no large corelation between N and insertion time
# The implementation must be correct for an O(logn) time

detach(delete_node_binomial)
```