empty.R

Preston

2020-05-23

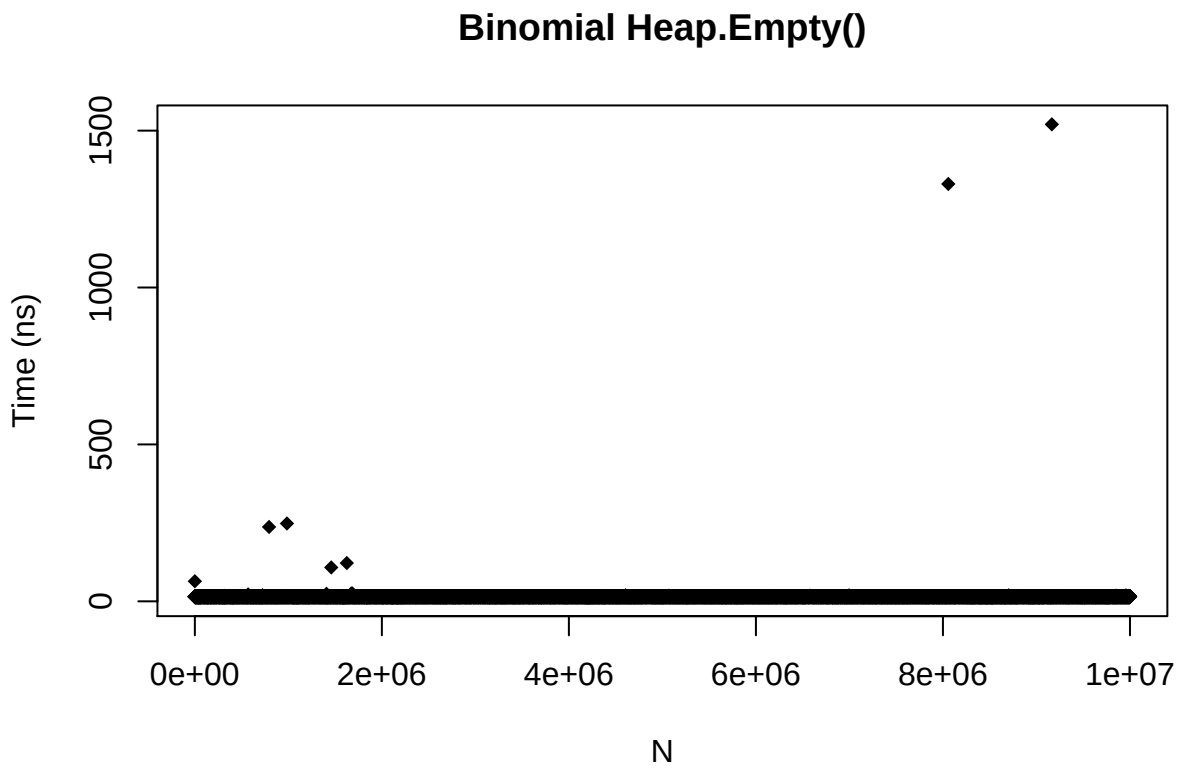
```
# Preston Dunton
# CS320 Honors Option
# May 20, 2020
# pdunton@rams.colostate.edu

# Should be O(1). Empty() asks if the heap is empty and returns a boolean, unlike
# clear(), which removes all elements from the heap

empty_binomial = read.csv("C:/Users/Preston/Desktop/CS320/HonOpt/empty_binomial.csv")
attach(empty_binomial)
```

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

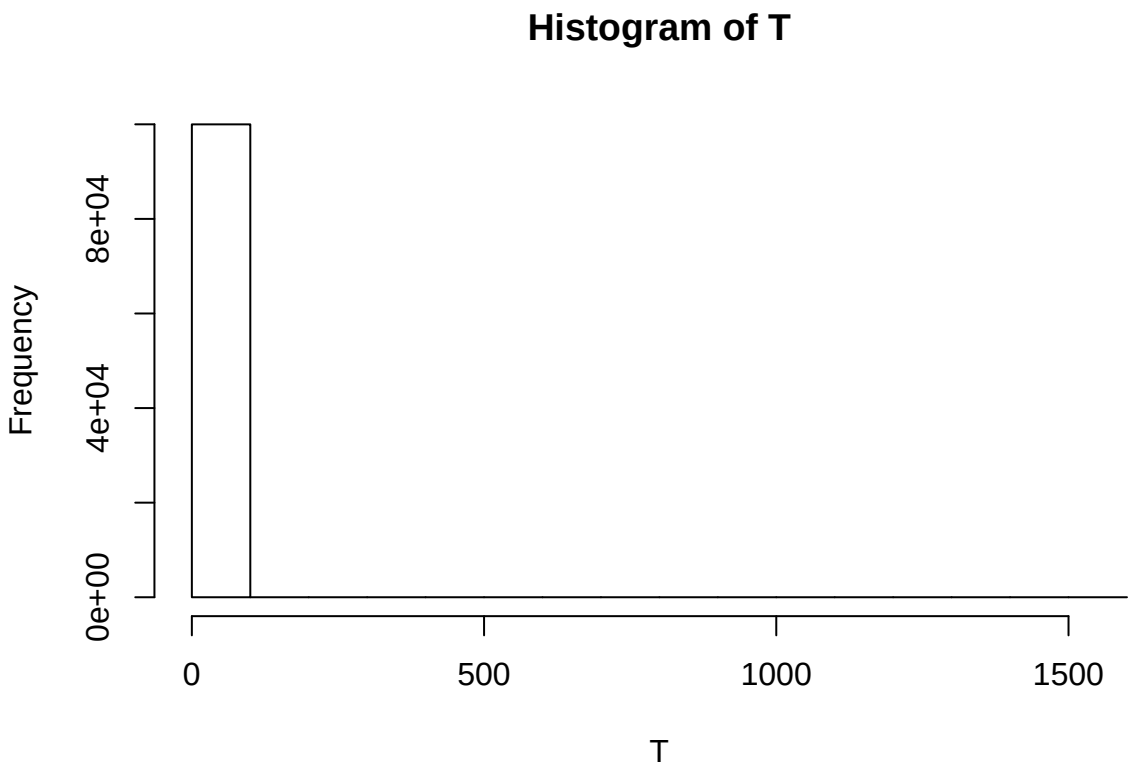
```
summary(T)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   13.00   14.00   15.00   14.64   15.00 1520.00
```

```
# min 13
# q1 14
# median 15
# mean 14.64
# q3 15
# max 1520

plot(N,T,pch=18,xlab="N",ylab="Time (ns)",main="Binomial Heap.Empty()")
```




```
hist(T,breaks=20)
```




```
# Let's try and remove some outliers

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

```
##   0%  10%  20%  30%  40%  50%  60%  70%  80%  90% 100%
##   13   14   14   14   14   15   15   15   15   15 1520
```

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

```
##  90%  91%  92%  93%  94%  95%  96%  97%  98%  99% 100%
##   15   15   15   15   15   15   16   16   16   16 1520
```

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

```
##   99% 99.1% 99.2% 99.3% 99.4% 99.5% 99.6% 99.7% 99.8% 99.9%  100%
##    16    16    16    16    16    16    16    16    16    16  1520
```

```
quantile(T,seq(0.999,1,0.0001))
```

```
##     99.9%    99.91%    99.92%    99.93%    99.94%    99.95%    99.96%    99.97%
##   16.0000   16.0000   16.0000   16.0000   16.0000   16.0000   16.0000   16.0003
##    99.98%    99.99%      100%
##   18.0000   23.0001 1520.0000
```

```
quantile(T,seq(0.9999,1,0.00001))
```

```
##     99.99%    99.991%    99.992%    99.993%    99.994%    99.995%    99.996%
##   23.00010   24.00009   25.00000   25.00273   64.00264  108.00070  122.00460
##    99.997%    99.998%    99.999%       100%
```

```
##  237.00033  248.02164 1330.00190 1520.00000
```

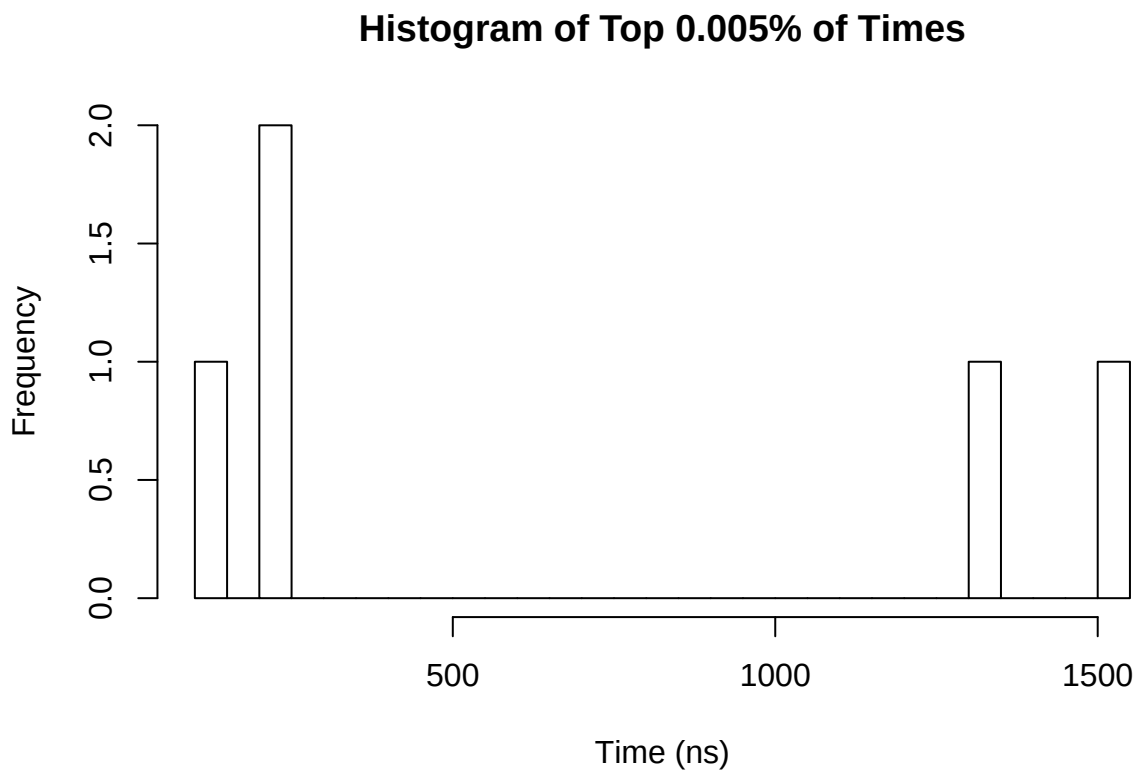
```
# Let's separate and anylze the top 0.005%

# Top 0.005%
sum(T>108) # There are 5 outliers
```

```
## [1] 5
```

```
summary(T[which(T>108)])
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   122.0   237.0   248.0   691.4  1330.0  1520.0
```

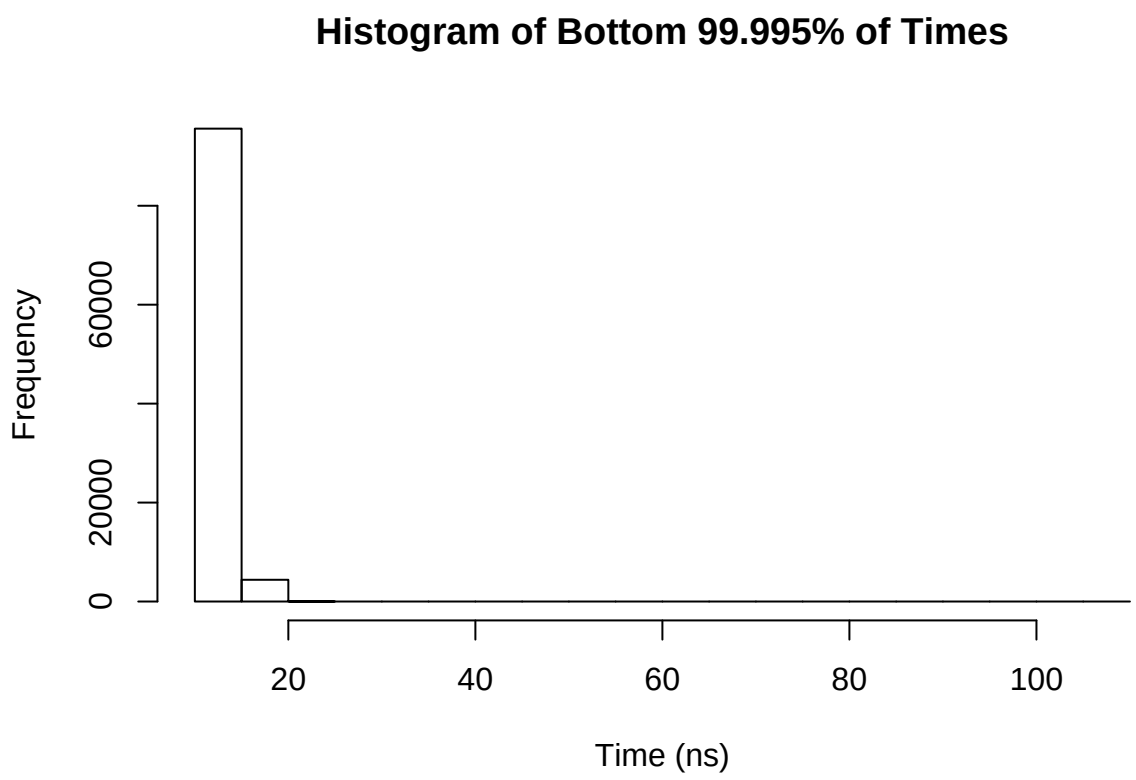
```
# min 122
# q1 237
# median 248
# mean 691.4
# q3 1330
# max 1520
hist(T[which(T>108)],main="Histogram of Top 0.005% of Times",xlab="Time (ns)",breaks=20)
```



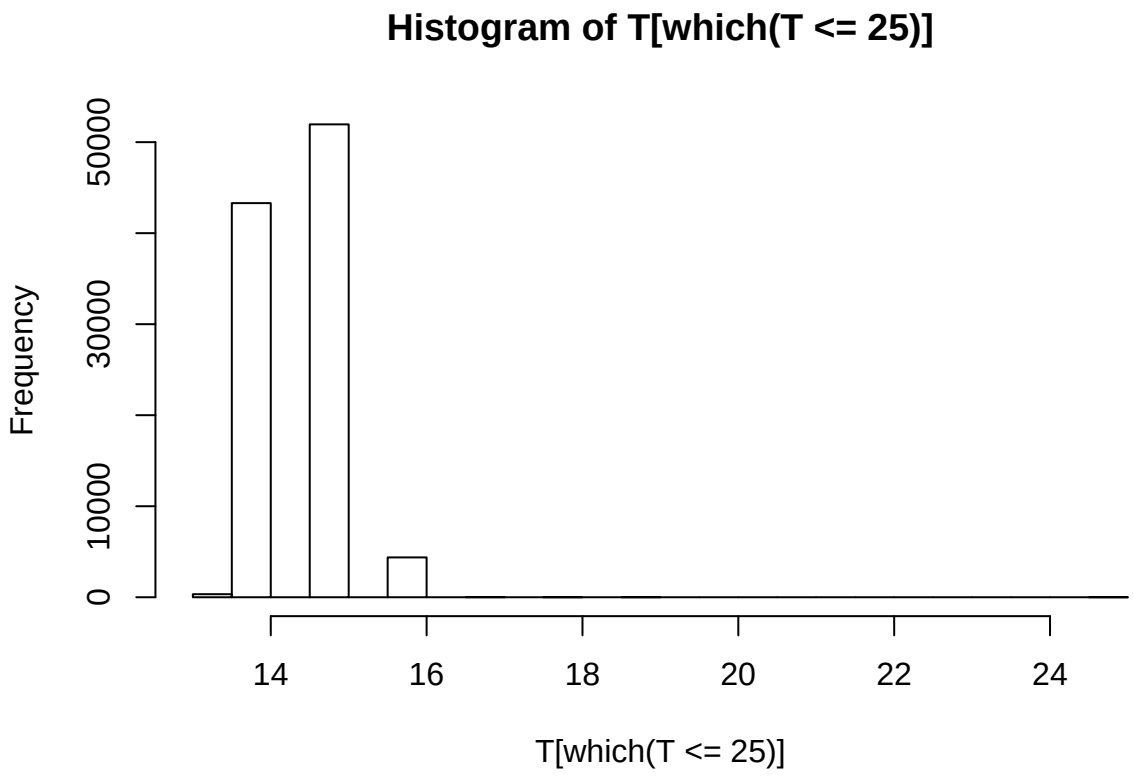

```
# Bottom 99.995%
summary(T[which(T<=108)])
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   13.00   14.00   15.00   14.61   15.00  108.00
```

```r
# min 13
# q1 14
# median 15
# mean 14.61
# q3 15
# max 108
hist(T[which(T<=108)],main="Histogram of Bottom 99.995% of Times",xlab="Time (ns)",breaks=20)
```




```r
# Let's get a better histogram to look at.
# Bottom 99.993%
hist(T[which(T<=25)],breaks=20)
```




```
# With some outliers (possible context switches or longer memory accesses)
# the time is on average 14.64.  Obviously O(1)

detach(empty_binomial)
```