heap_union.R

Preston

2020-05-23

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

# Heap_Union() in a binomial heap should be O(log(n+m))
# where the two heaps have a size of n and m respectively

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

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

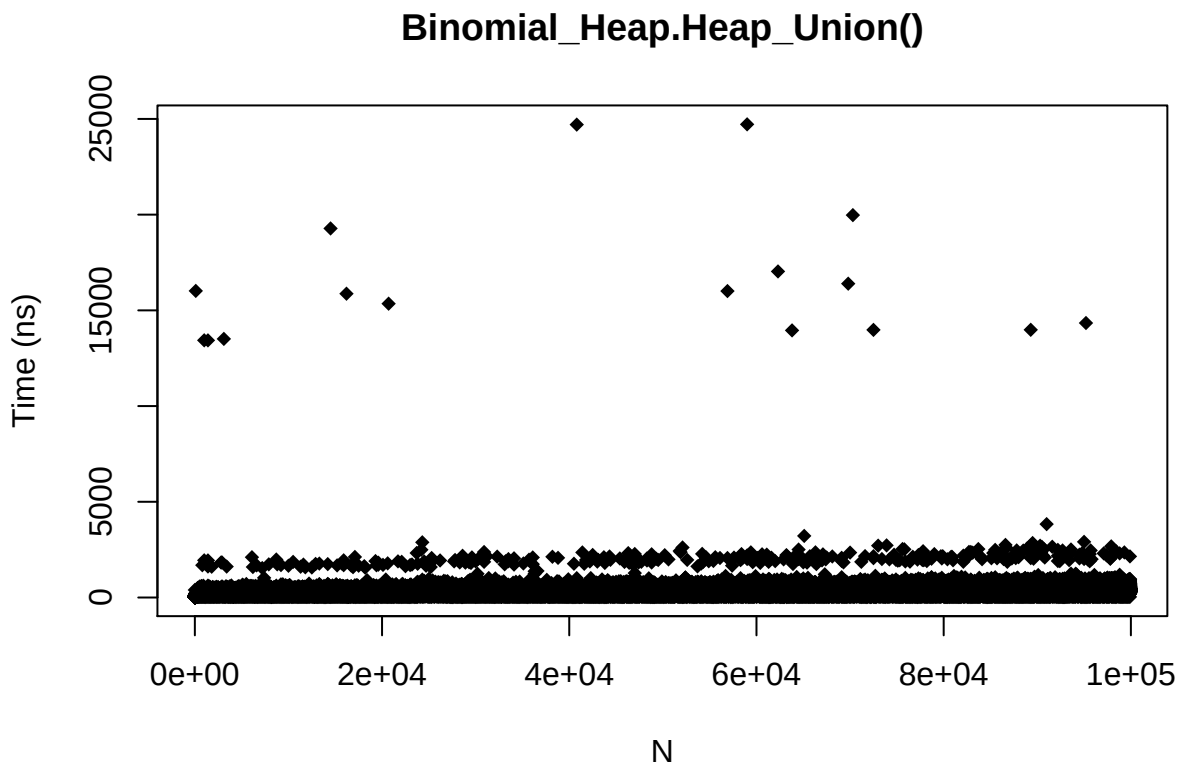
```
# Note: this dataset is larger because it records for all combinations of n and m
length(T)
```

```
## [1] 1000000
```

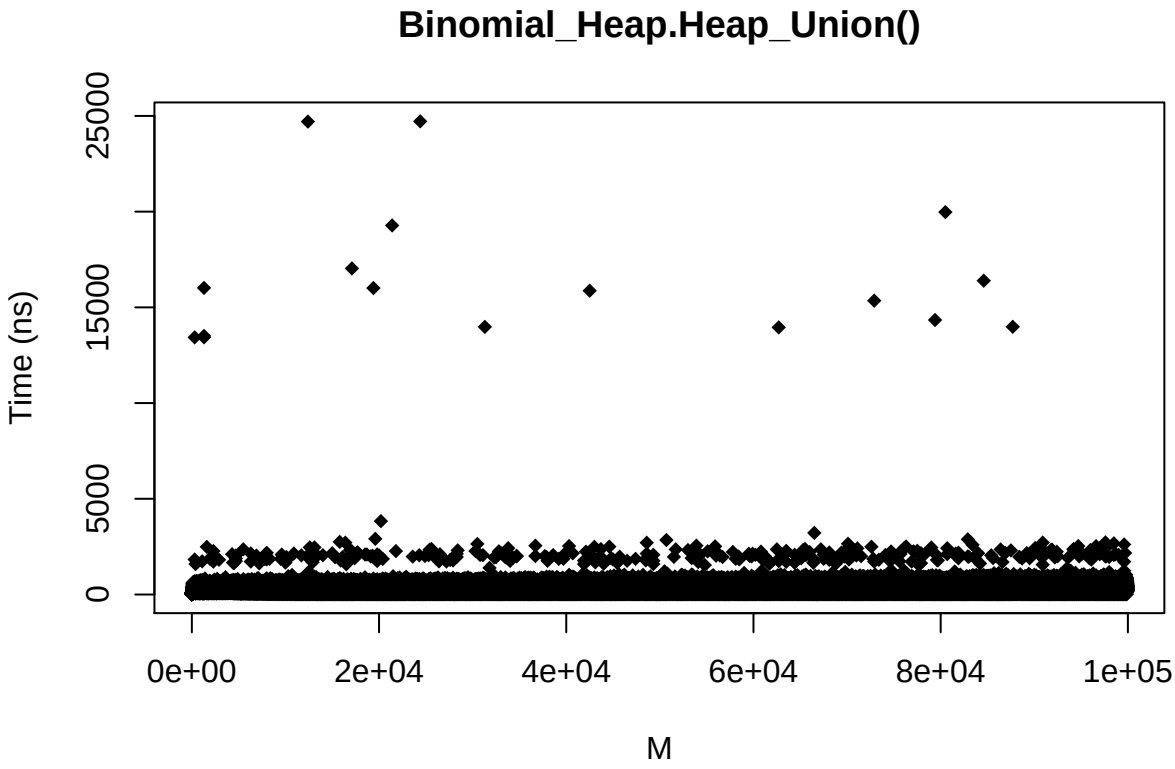
```
# We will also need a variable for n+m
NplusM = N+M

summary(T)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    14.0   220.0   274.0   292.3   341.0 24717.0
```

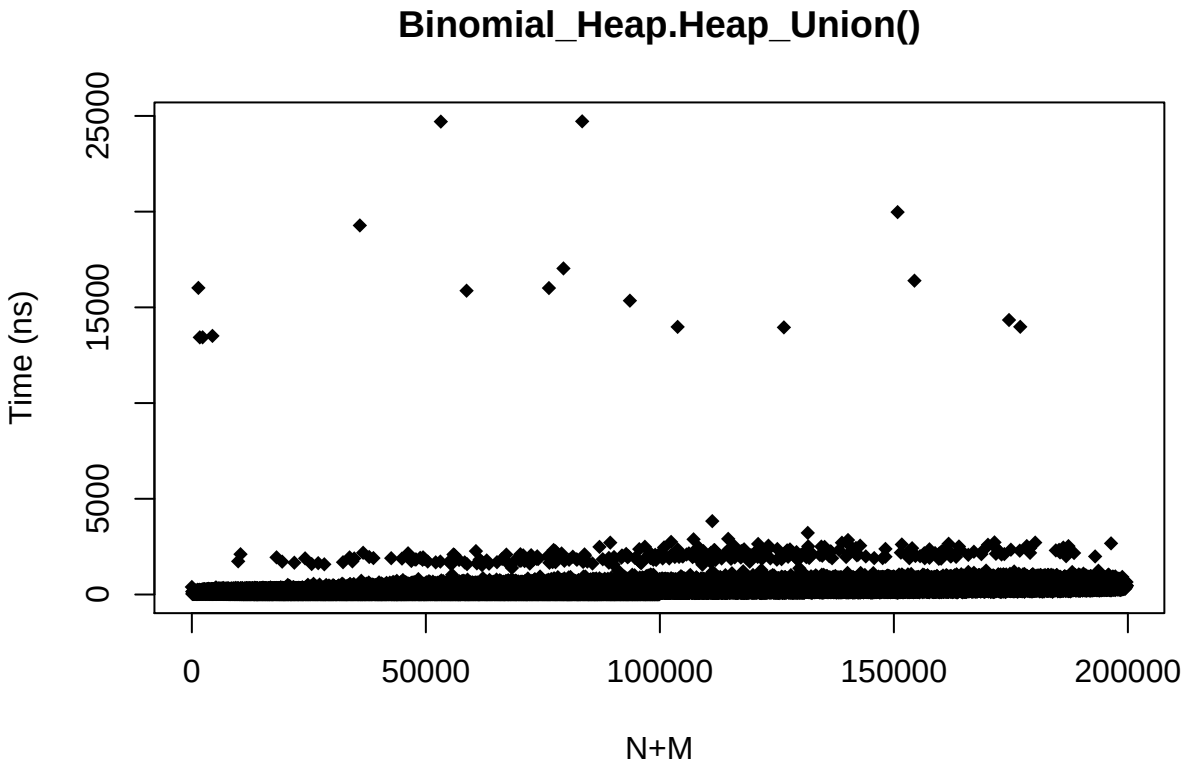
```
# min 14
# q1 220
# median 274
# mean 292
# q3 341
# max 24717
plot(N,T,pch=18,xlab="N",ylab="Time (ns)",main="Binomial_Heap.Heap_Union()")
```



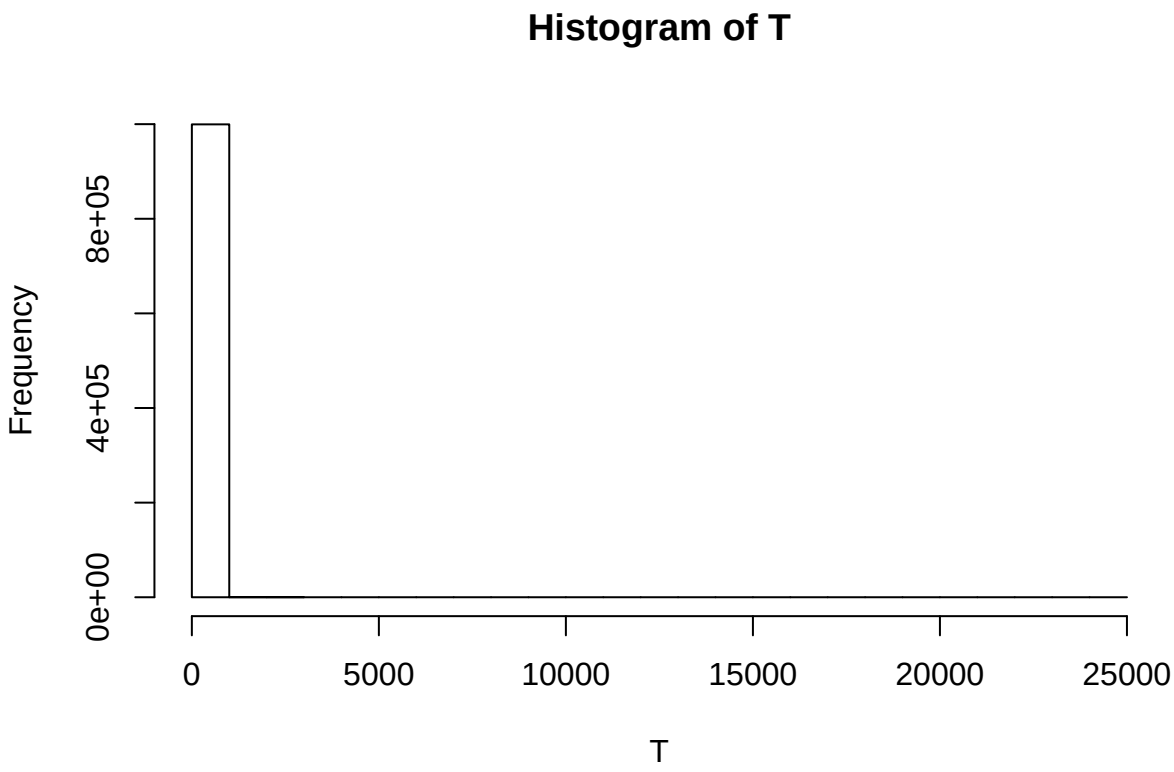

```
plot(M,T,pch=18,xlab="M",ylab="Time (ns)",main="Binomial_Heap.Heap_Union()")
```




```
plot(NplusM,T,pch=18,xlab="N+M",ylab="Time (ns)",main="Binomial_Heap.Heap_Union()")
```




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




```
# 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%
##    14   178   208   231   252   274   297   324   362   425 24717
```

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

```
##   90%   91%   92%   93%   94%   95%   96%   97%   98%   99%  100%
##   425   434   446   458   472   488   510   540   586   665 24717
```

```
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%
##   665   676   687   701   716   733   753   778   813   872 24717
```

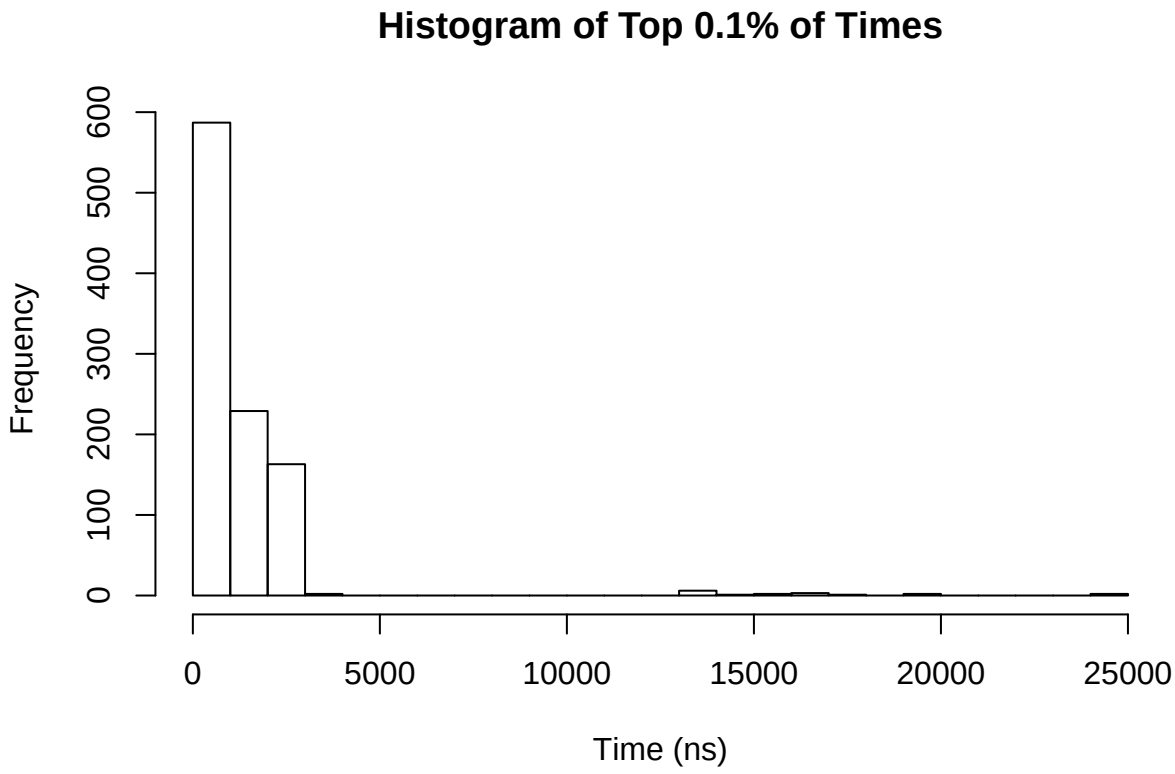
```
# Let's separate the top 0.1% and analyze

# Top 0.1%
sum(T>872) # There are 998 outliers
```

```
## [1] 998
```

```
summary(T[which(T>872)])
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   873.0   904.0   950.5  1545.3  1861.8 24717.0
```

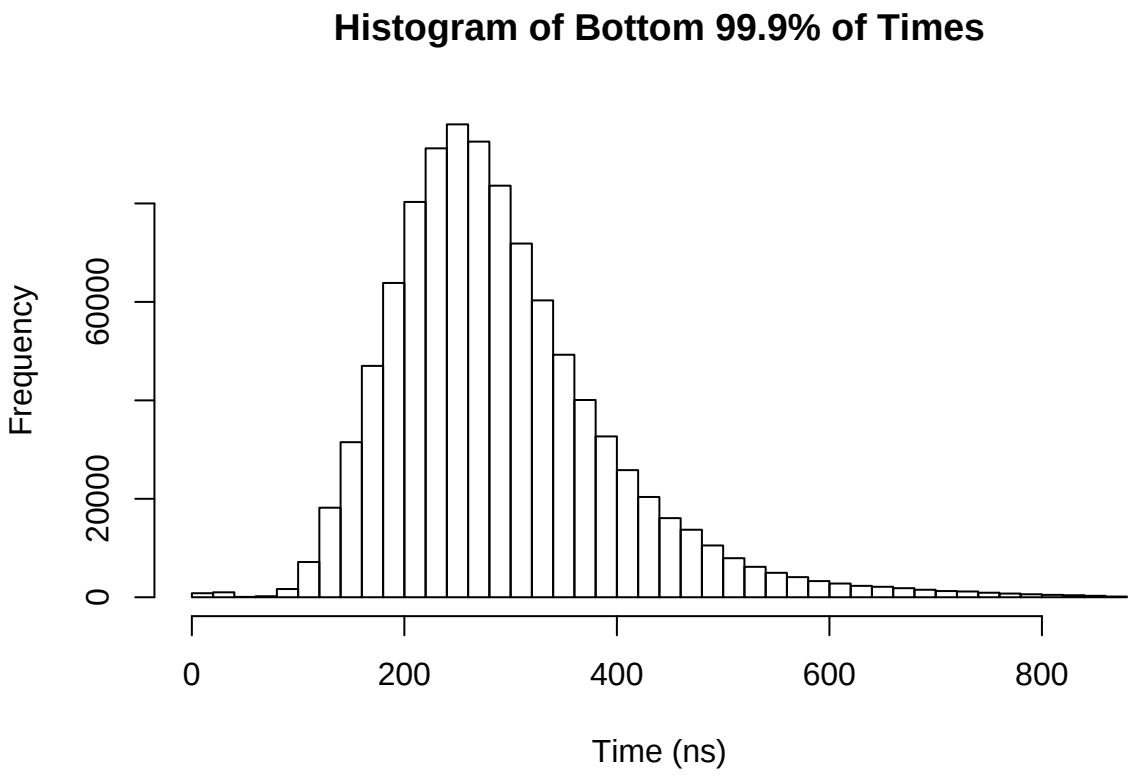
```r
# min 873
# q1 904
# median 950.5
# mean 1545.3
# q3 1861.8
# max 24717
hist(T[which(T>872)],main="Histogram of Top 0.1% of Times",xlab="Time (ns)",breaks=30)
```



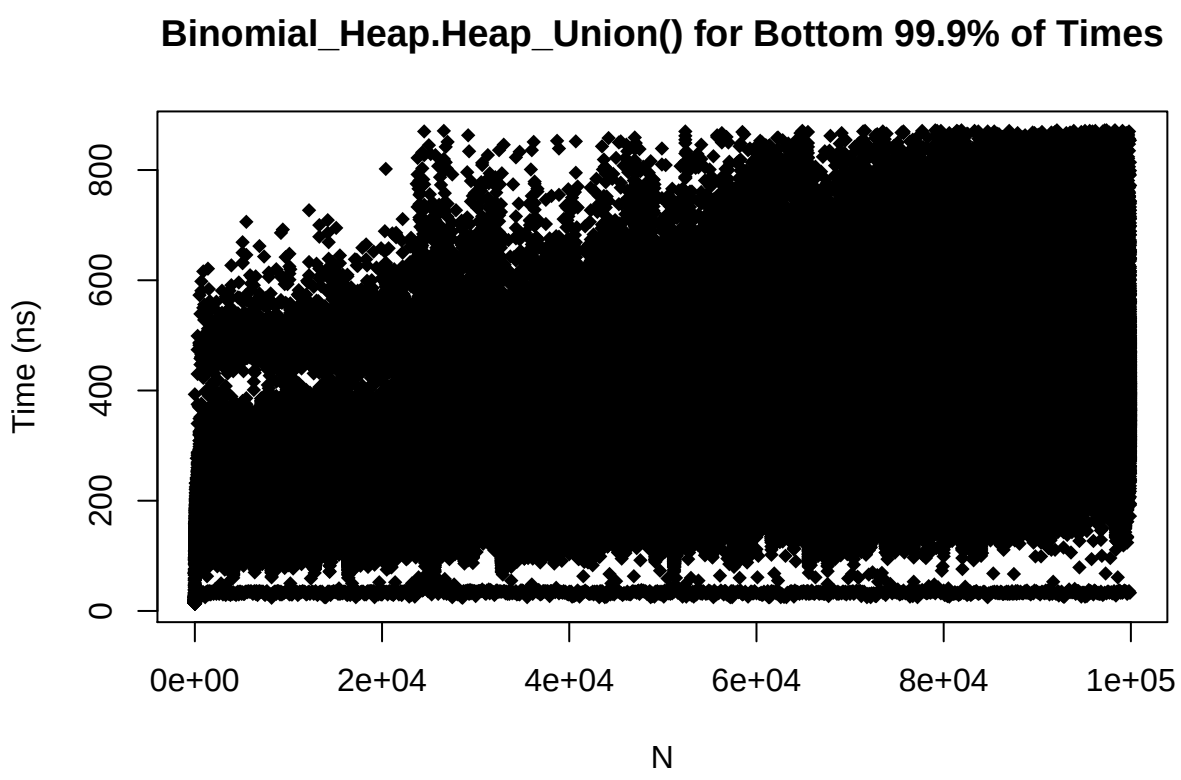

```r
# Bottom 99.9%
summary(T[which(T<=872)])
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      14     220     273     291     341     872
```

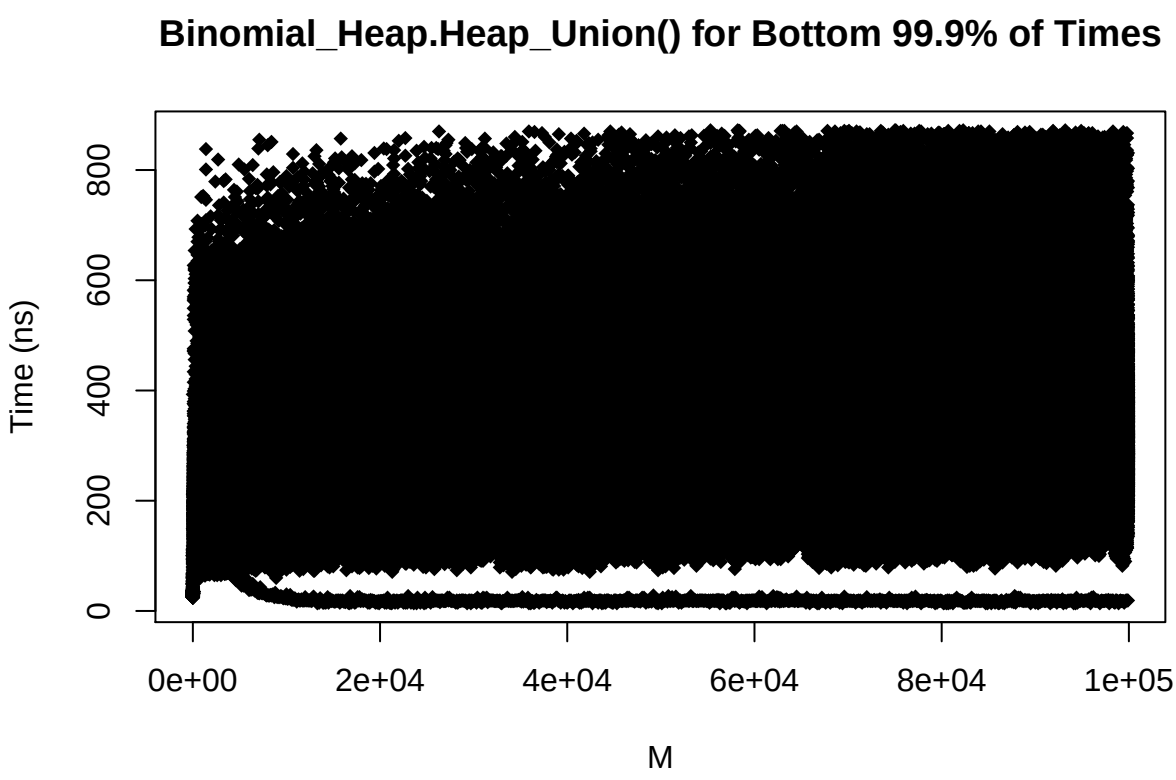
```r
# min 14
# q1 220
# median 273
# mean 291
# q3 341
# max 872
hist(T[which(T<=872)],main="Histogram of Bottom 99.9% of Times",xlab="Time (ns)",breaks=40)
```



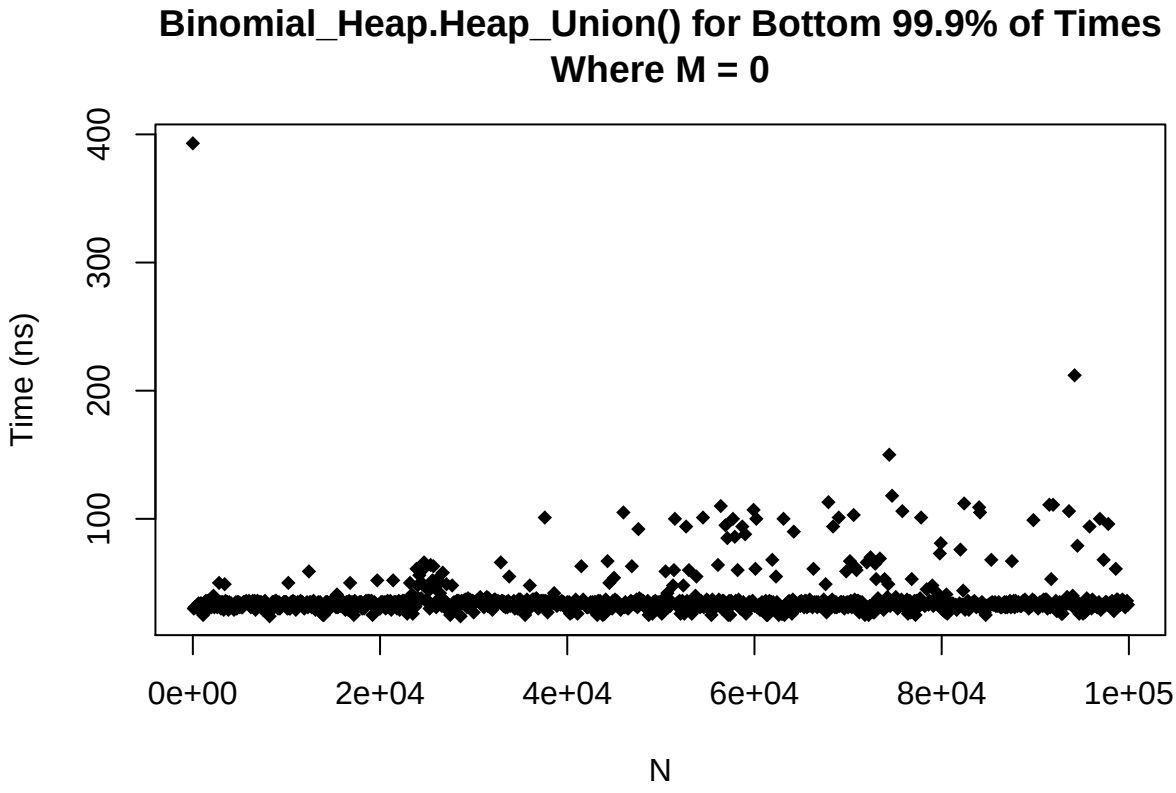

```
# Let's look at the scatterplots again without the outliers
plot(N[which(T<=872)],T[which(T<=872)],pch=18,xlab="N",ylab="Time (ns)",main="Binomial_Heap.Heap_Union(
```



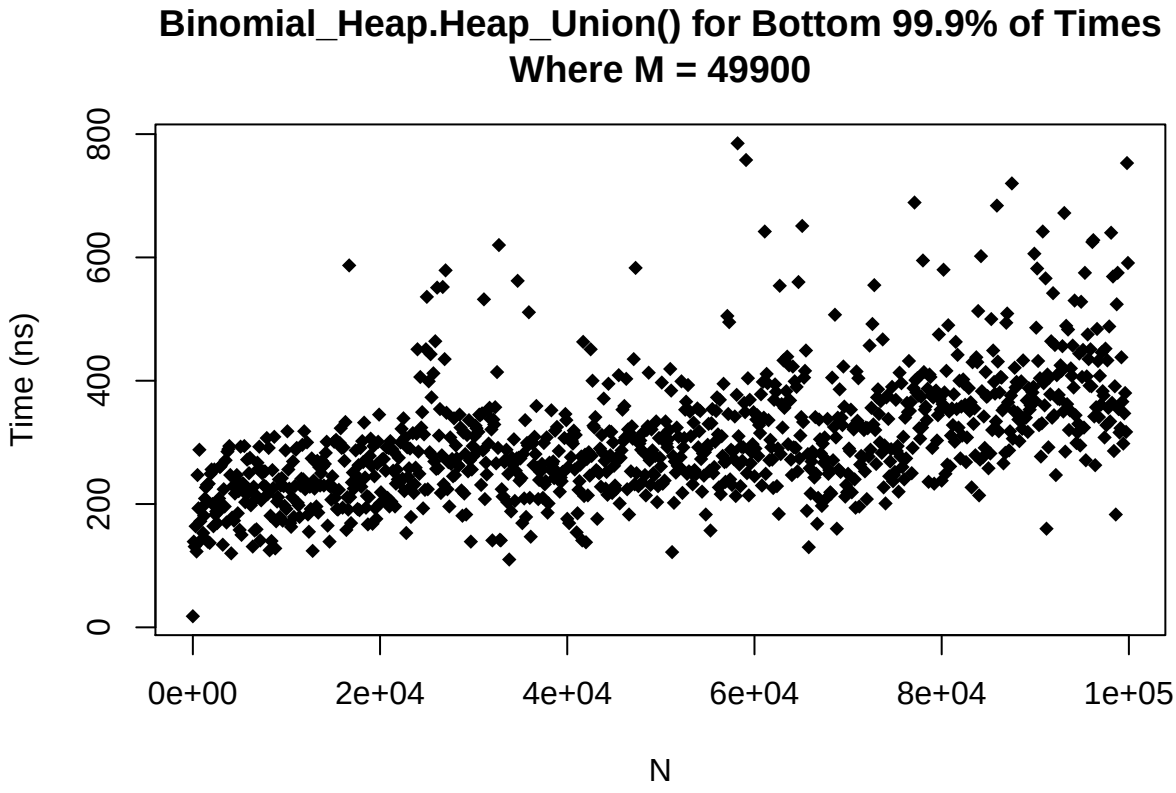

```
plot(M[which(T<=872)],T[which(T<=872)],pch=18,xlab="M",ylab="Time (ns)",main="Binomial_Heap.Heap_Union(
```



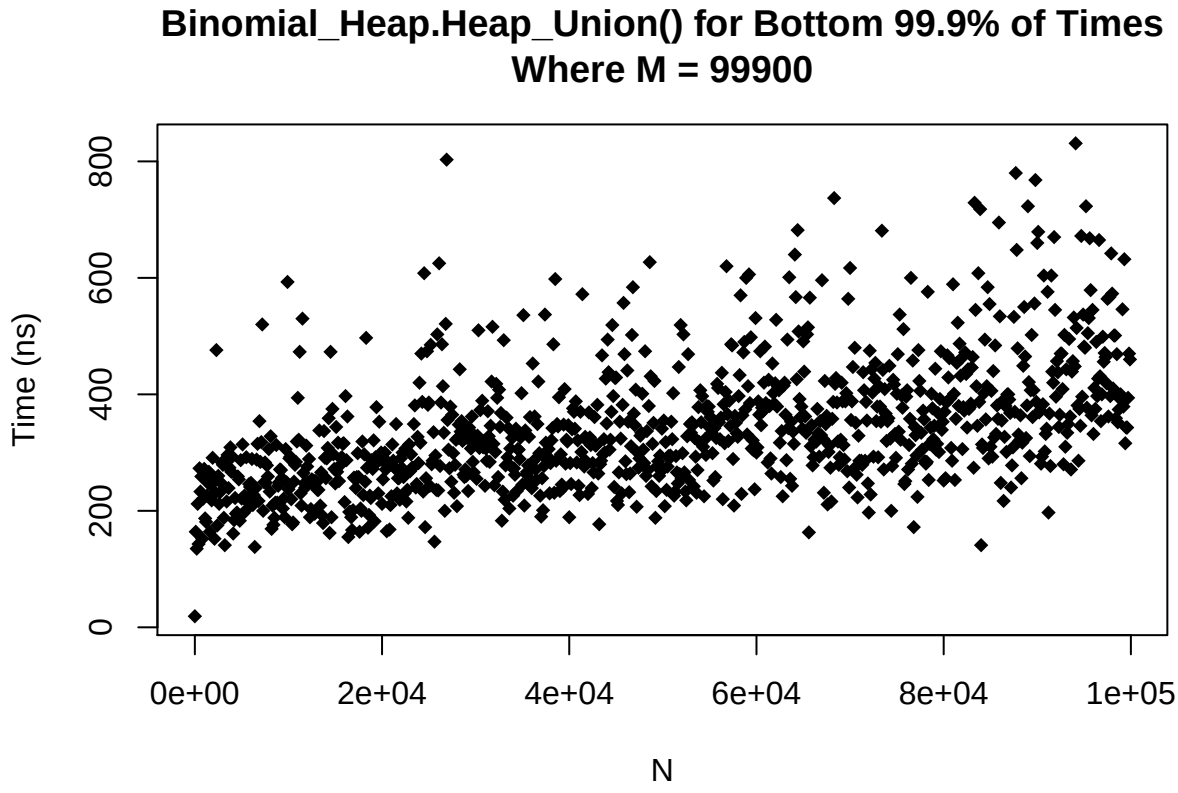

```
# These are a mess!  This is because there are multiple T values for a single N or M value.
# Let's hold N and M constant and retry.
# N and M take on the same uniform distribution between 0 and 100000.
# Let's use the start, middle, and end of that range to create scatterplots

plot(N[which(T<=872 & M==0)],T[which(T<=872 & M==0)],pch=18,xlab="N",ylab="Time (ns)",main="Binomial_Hea
```



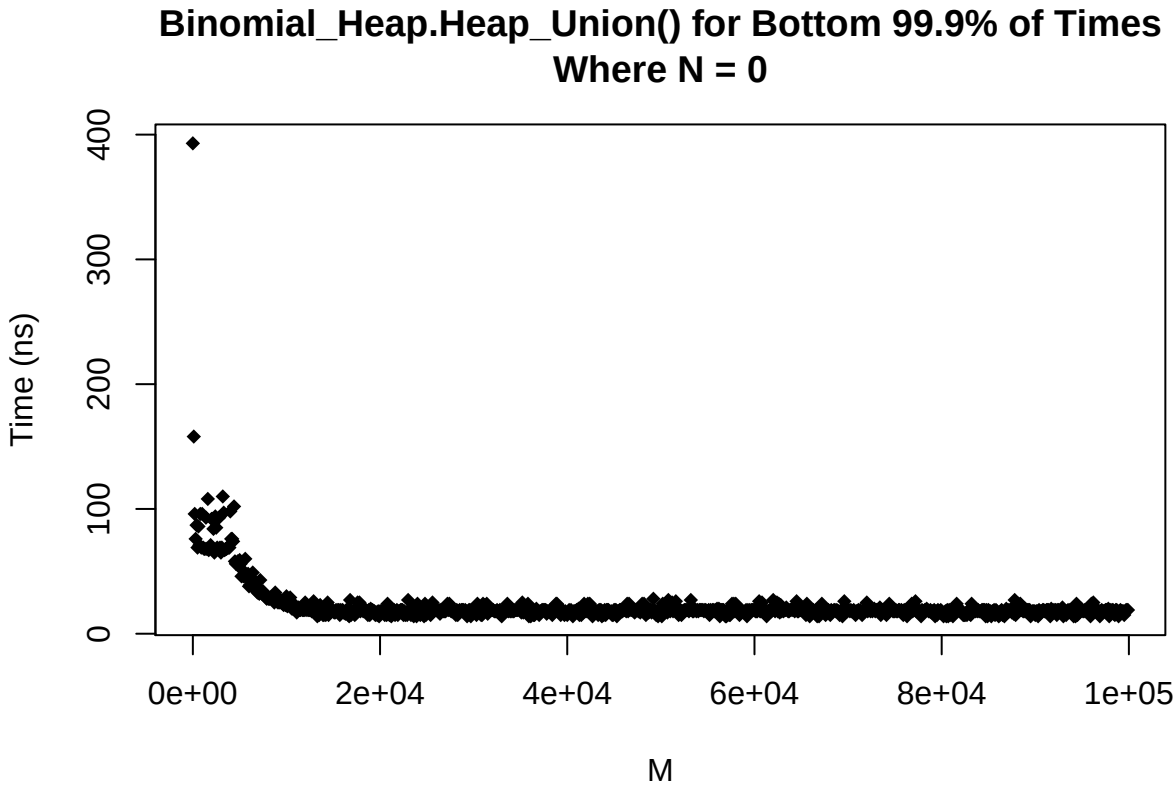

```
plot(N[which(T<=872 & M==49900)],T[which(T<=872 & M==49900)],pch=18,xlab="N",ylab="Time (ns)",main="Bin
```



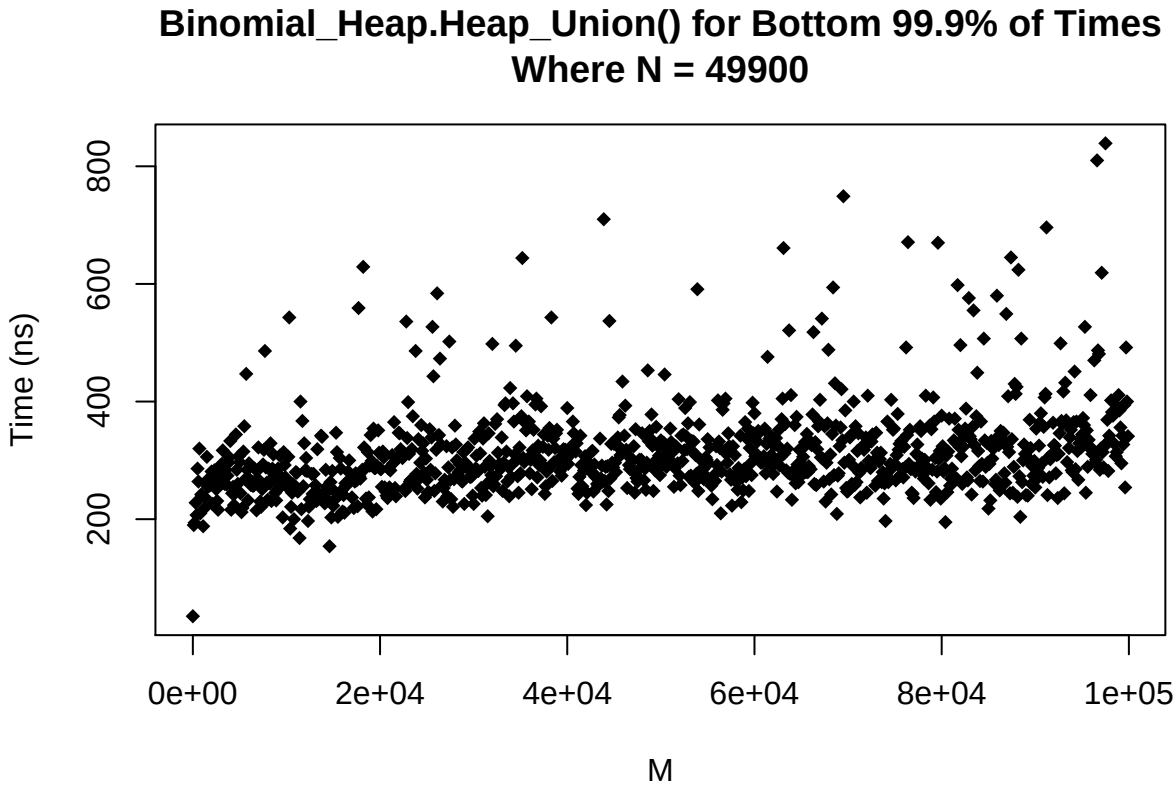

```
plot(N[which(T<=872 & M==99900)],T[which(T<=872 & M==99900)],pch=18,xlab="N",ylab="Time (ns)",main="Bin
```



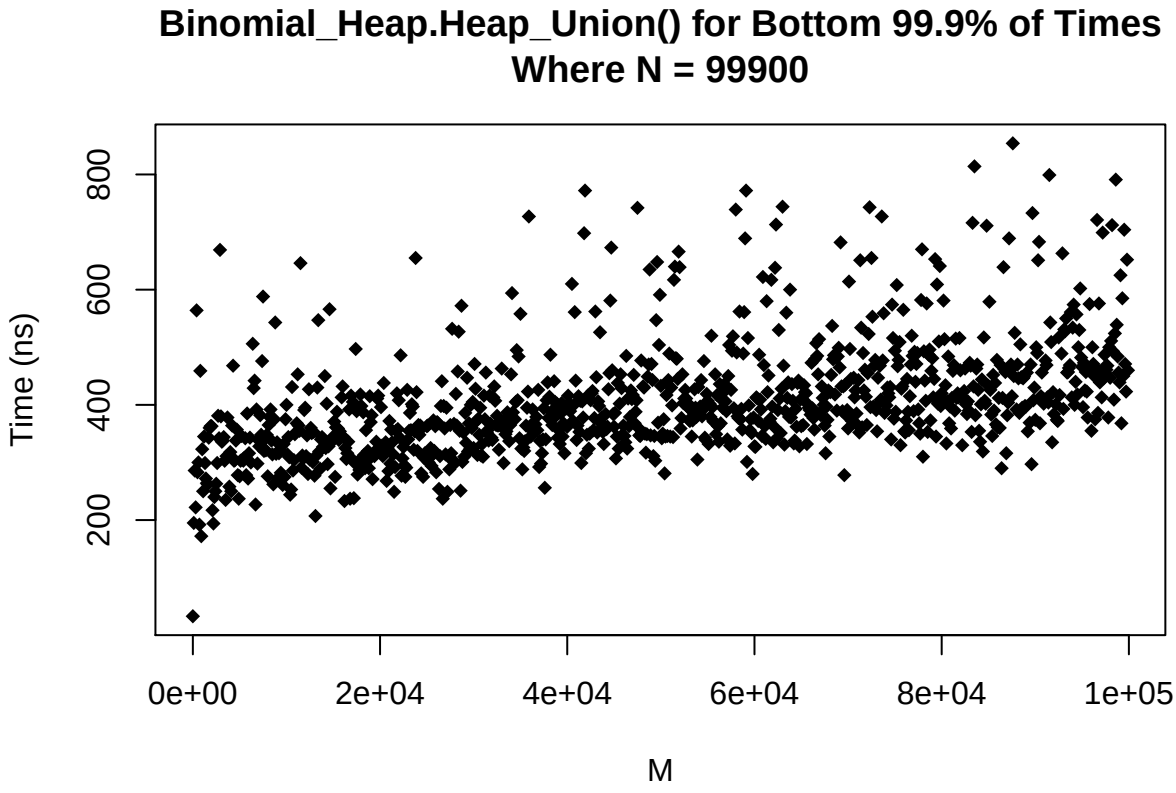

```
plot(M[which(T<=872 & N==0)],T[which(T<=872 & N==0)],pch=18,xlab="M",ylab="Time (ns)",main="Binomial_Hea
```



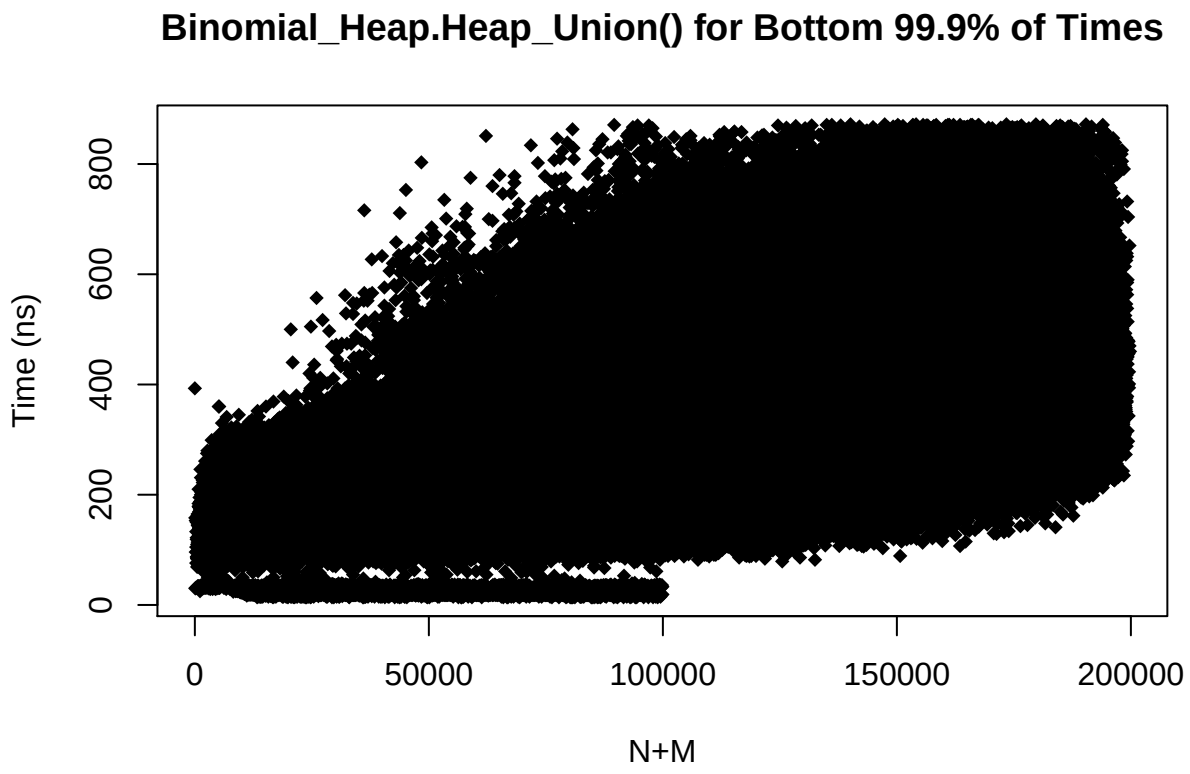

```
# Interesting trend here.  The times start high and decrease.
plot(M[which(T<=872 & N==49900)],T[which(T<=872 & N==49900)],pch=18,xlab="M",ylab="Time (ns)",main="Bin
```




```
plot(M[which(T<=872 & N==99900)],T[which(T<=872 & N==99900)],pch=18,xlab="M",ylab="Time (ns)",main="Bin
```




```
# Since we know that the times should be based on N+M, let's stop all these scatterplots
# and use our N+M variable

plot(NplusM[which(T<=872)],T[which(T<=872)],pch=18,xlab="N+M",ylab="Time (ns)",main="Binomial_Heap.Heap_
```




```
# This is also a mess.  Let's do multiple regression to take a look
# at the relationship between N+M and T using numbers.

cor(NplusM,T) # Lower correlation of 0.4497216.  Good sign for O(log(N+M))
```

```
## [1] 0.4497216
```

```
lm(T~N+M)
```

```
##
## Call:
## lm(formula = T ~ N + M)
##
## Coefficients:
## (Intercept)            N            M
##   1.482e+02    1.969e-03    9.156e-04
```

```
lm(T~NplusM)
```

```
##
## Call:
## lm(formula = T ~ NplusM)
##
## Coefficients:
## (Intercept)       NplusM
##   1.482e+02    1.442e-03
```

```
# All coeffecients are very small, which suggests that the relationship is not linear
# as we have seen in operations we know to be linear in time.
```

```
# Because we see positive coefficients and a general increase in time as N and M increase,
# we can assume that our implementation is correct for O(log(n+m))

detach(heap_union_binomial)
```