assignment_operator.R

Preston

2020-05-23

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

# The assignment operator =  of a binomial heap should be O(n+m)
# where the assignment is heap_of_size_n = heap_of_size_m;

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

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

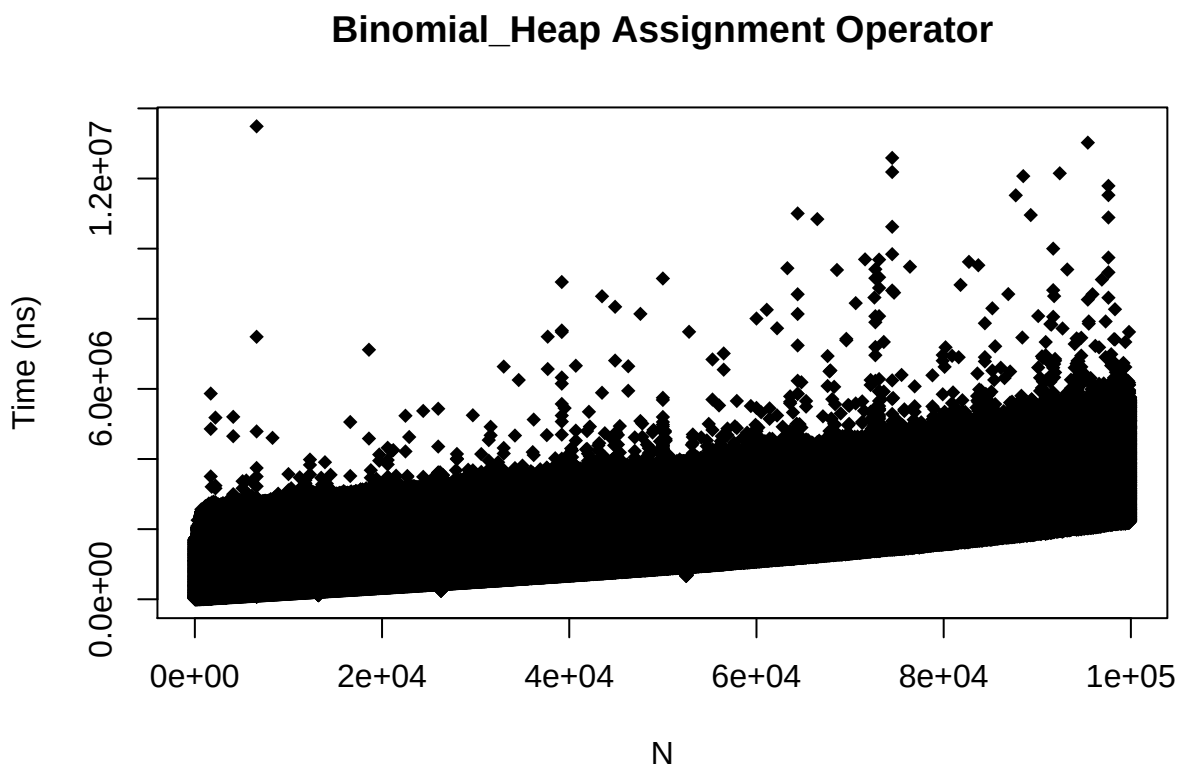
```
# Note: this dataset is larger because it records for all combinations of n and m
length(T)
```

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

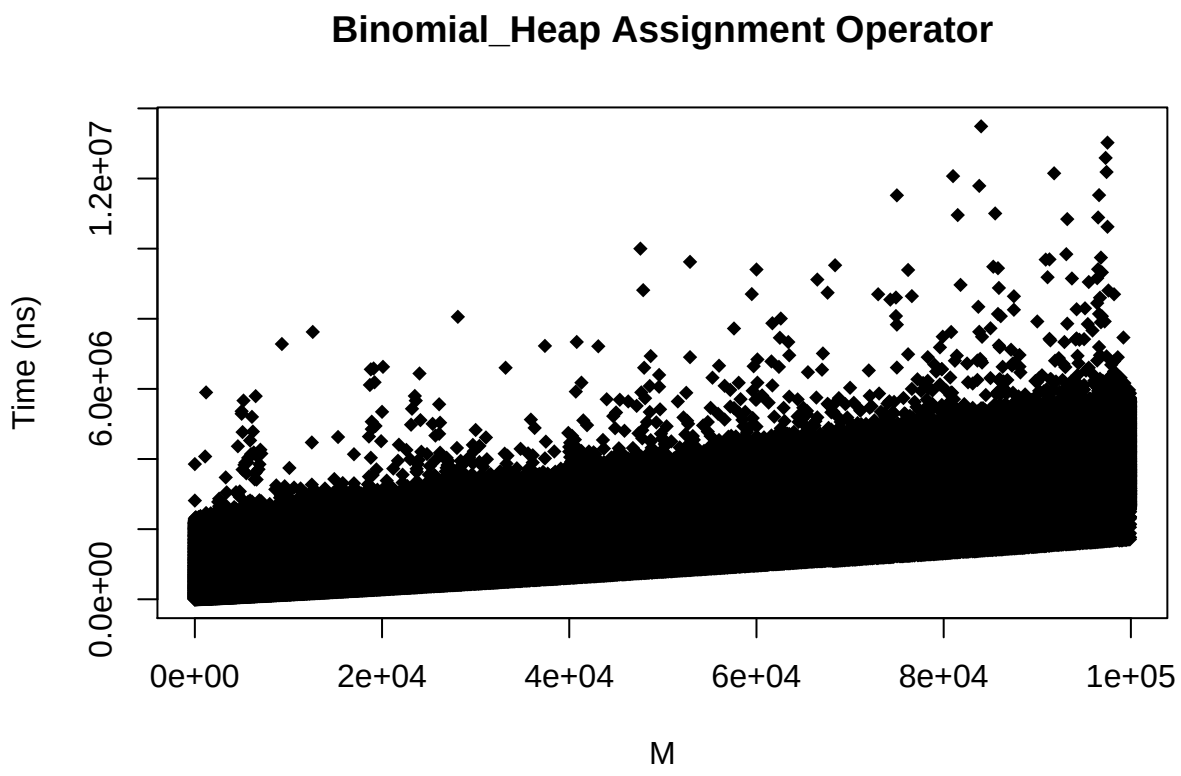
```
# We will also need a variable for n+m
NplusM = N+M

summary(T)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
##      767  1528234  2331850  2399332  3171077 13487930
```

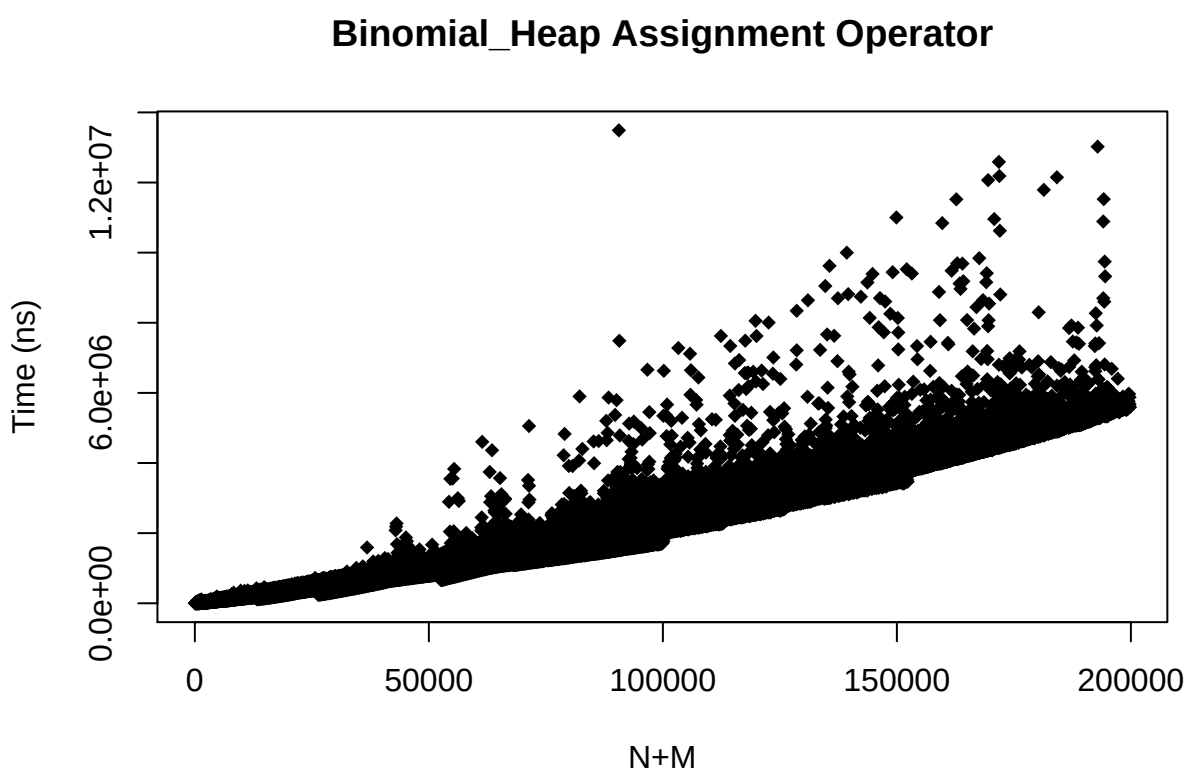
```
# min 767
# q1 1528234
# median 2331850
# mean 2399332
# q3 3171077
# max 13487930
plot(N,T,pch=18,xlab="N",ylab="Time (ns)",main="Binomial_Heap Assignment Operator")
```



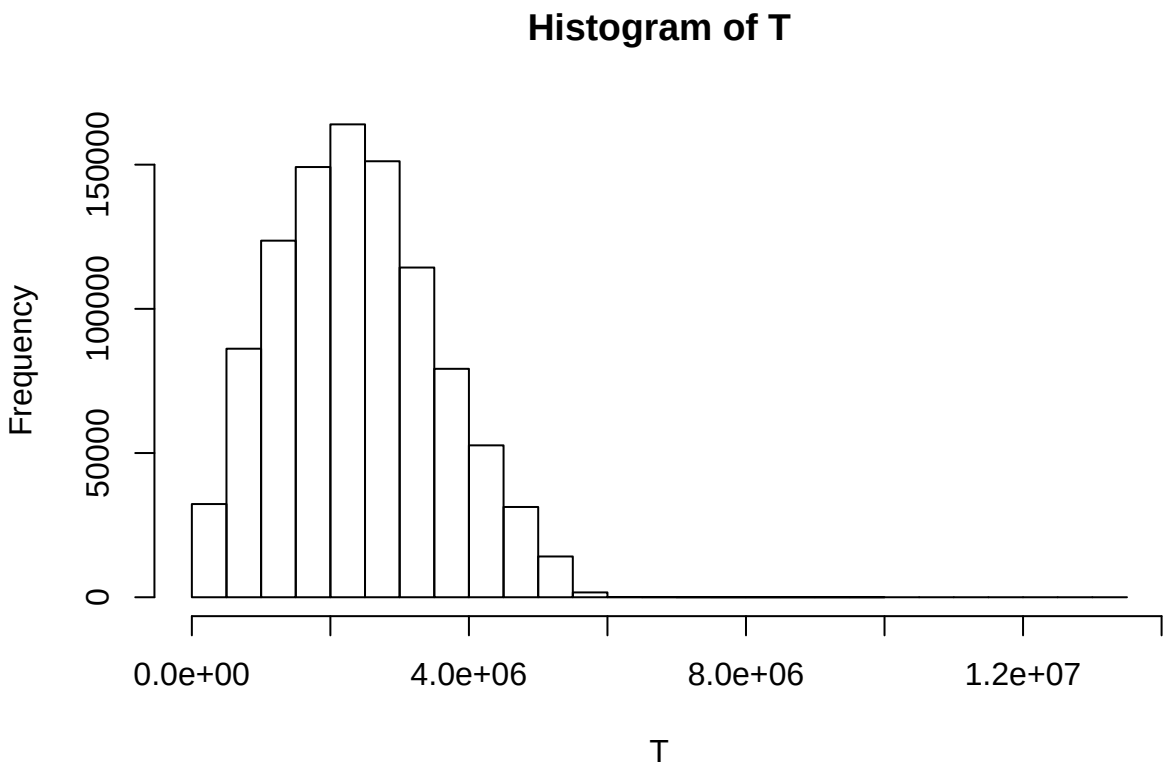

```
plot(M,T,pch=18,xlab="M",ylab="Time (ns)",main="Binomial_Heap Assignment Operator")
```



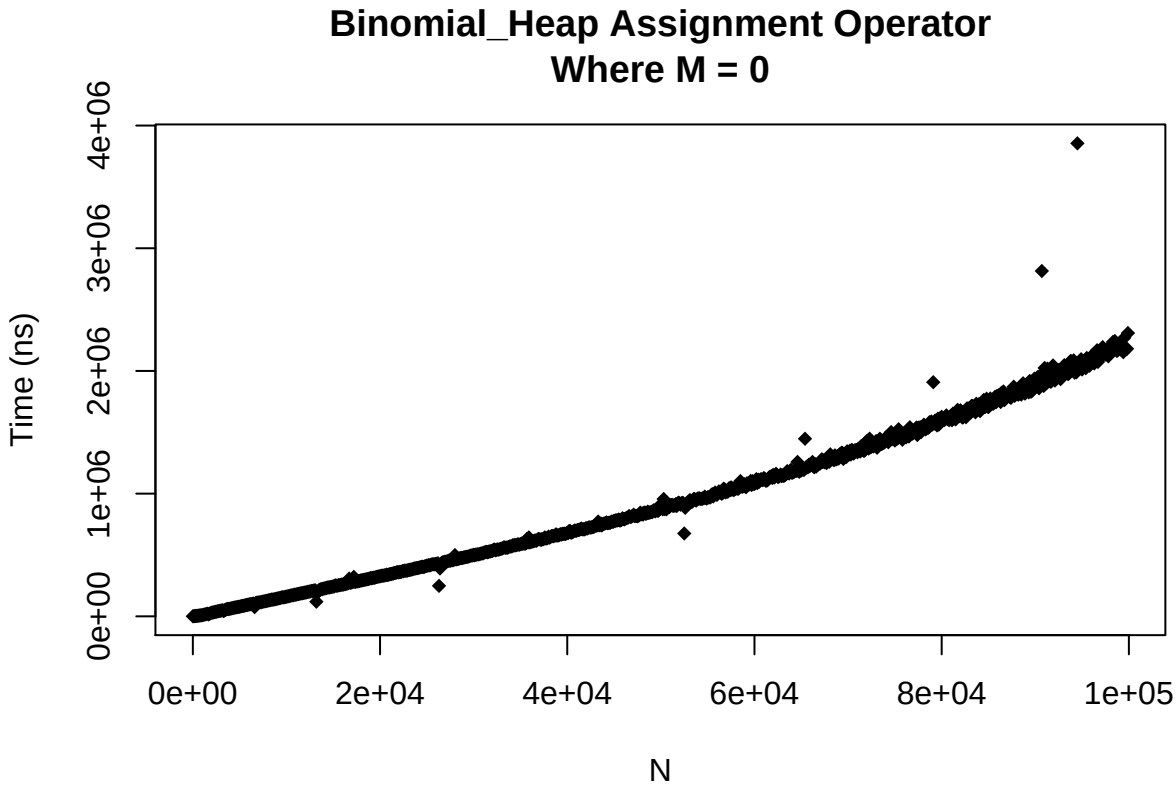

```
plot(NplusM,T,pch=18,xlab="N+M",ylab="Time (ns)",main="Binomial_Heap Assignment Operator")
```



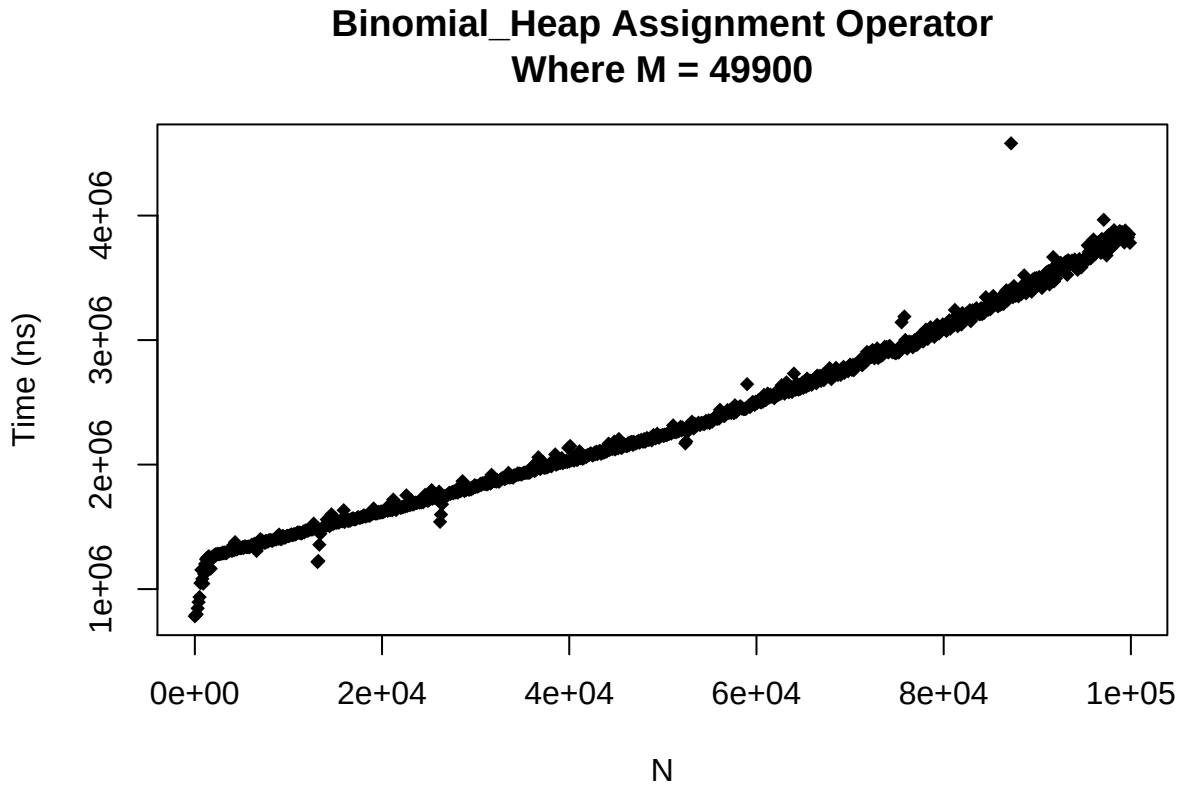

```
# It's interesting to see how this last chart clusters the data closer than the others.
# There is also some strange edges at the bottom of trend curve.
# This could maybe be the same patterns observed in the clear().
# Both the assignment operator and clear() make use of the delete_tree() internal function,
# which could be the cause of this pattern.

hist(T,breaks=30)
```



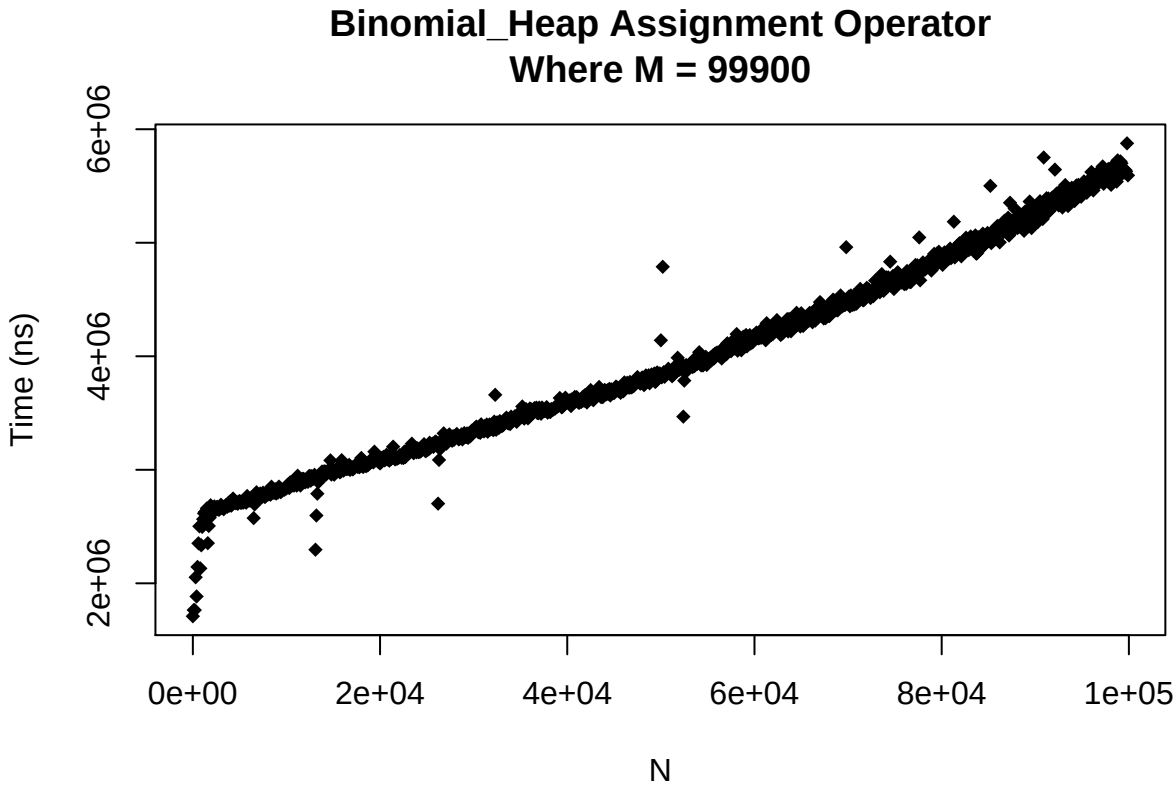

```
# A normal distribution.  Something unexpected for an O(n+m) function.  Other linear functions
# we have seen, such as the copy constructor and clear() demonstrate a near uniform distribution.

# These scatterplots 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(M==0)],T[which(M==0)],pch=18,xlab="N",ylab="Time (ns)",main="Binomial_Heap Assignment Opera
```



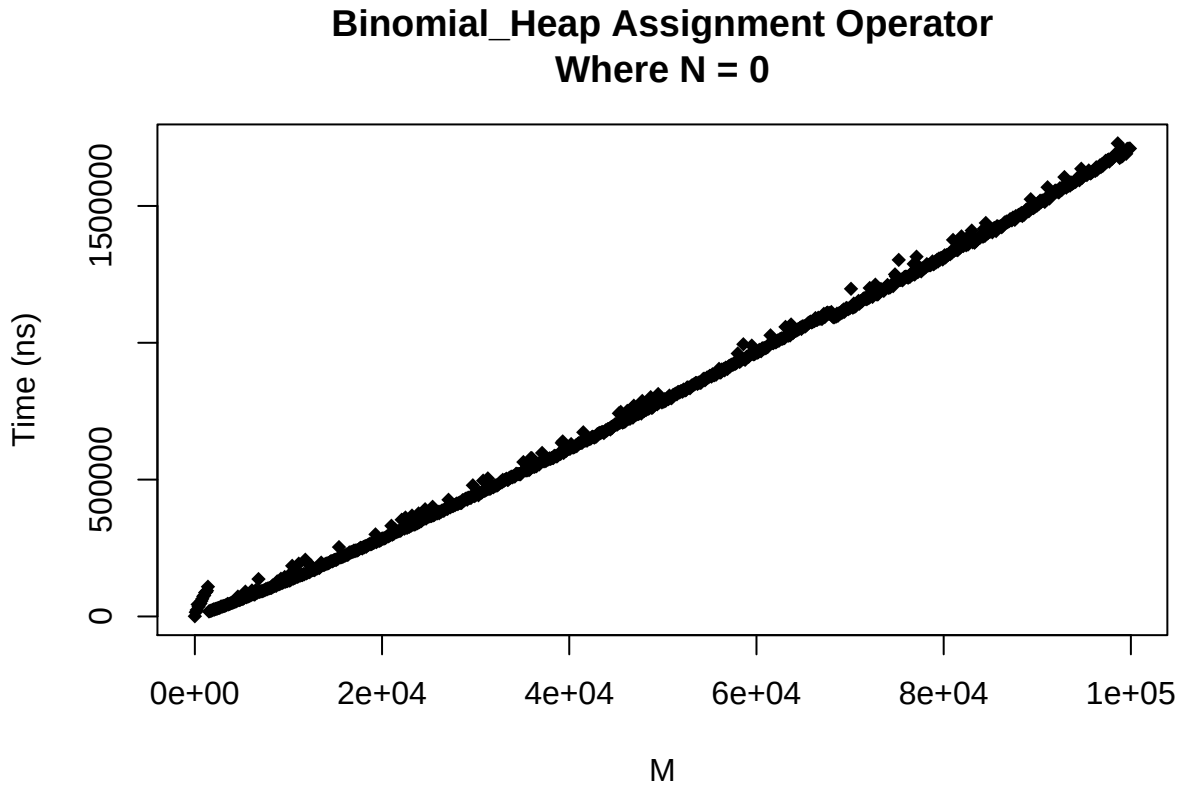

```
plot(N[which(M==49900)],T[which(M==49900)],pch=18,xlab="N",ylab="Time (ns)",main="Binomial_Heap Assignme
```



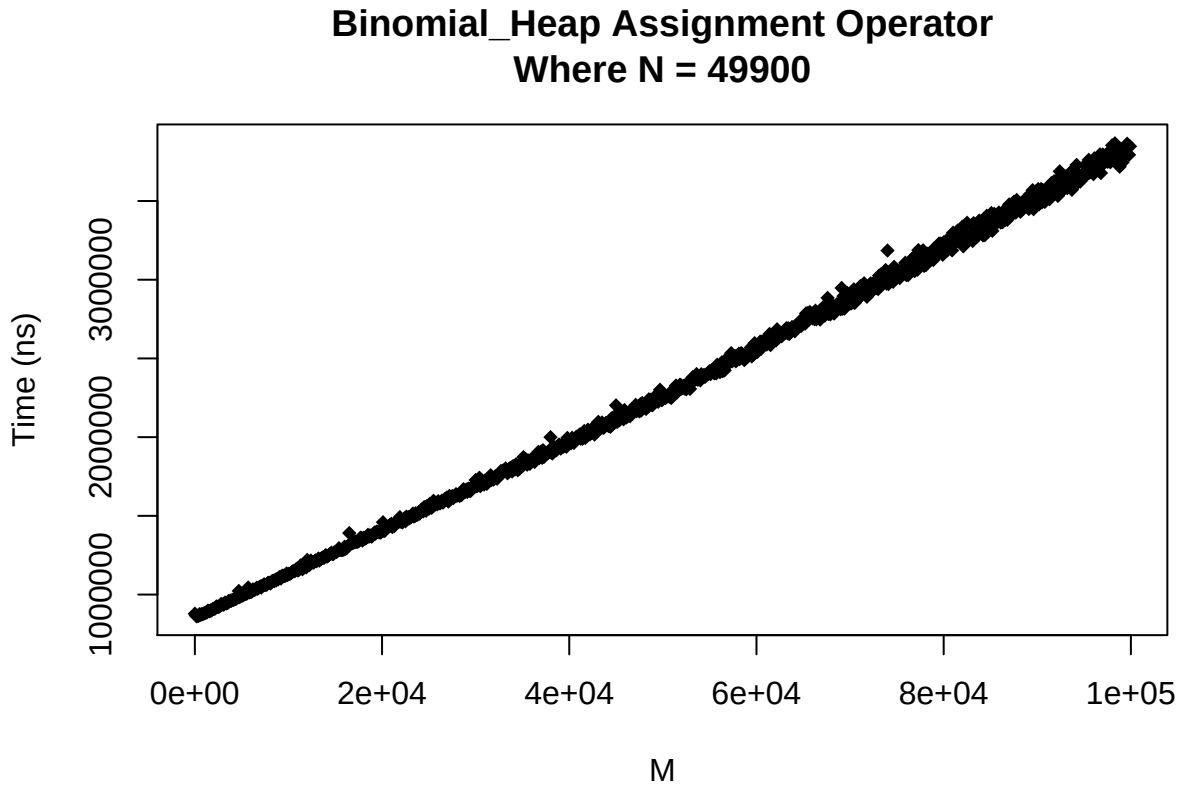

```
plot(N[which(M==99900)],T[which(M==99900)],pch=18,xlab="N",ylab="Time (ns)",main="Binomial_Heap Assignme
```



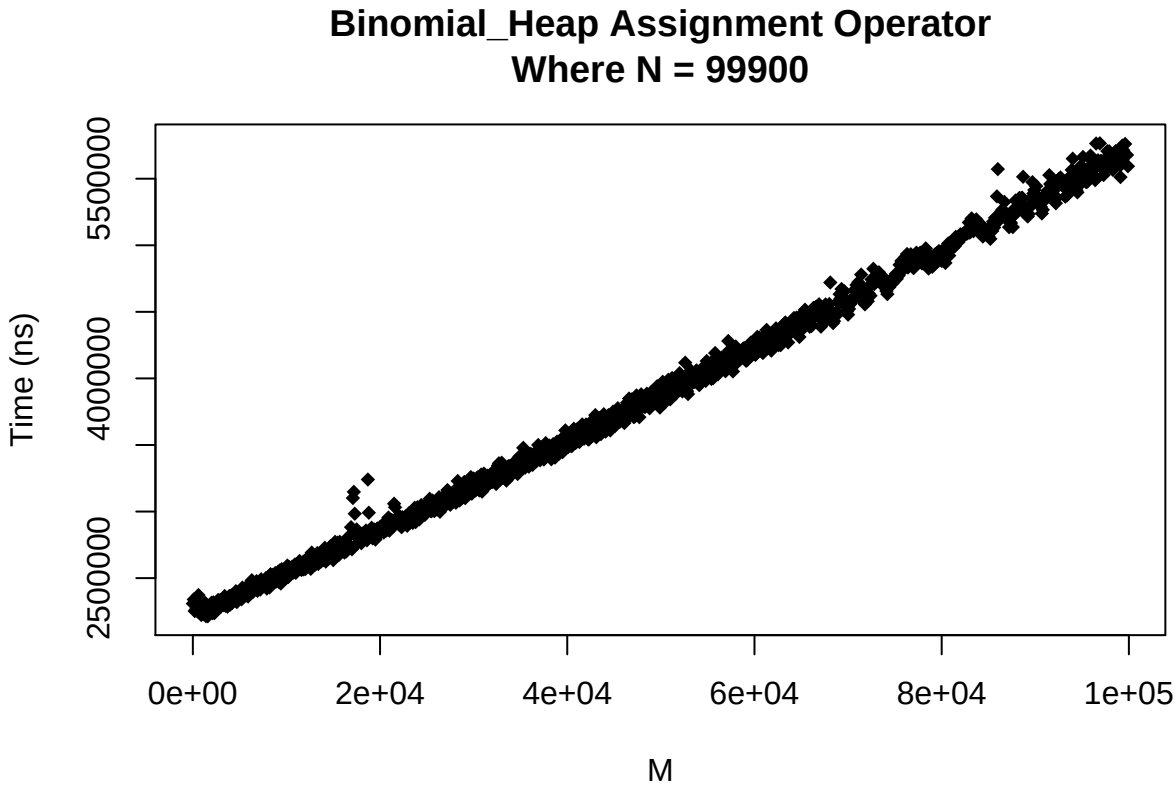

```
plot(M[which(N==0)],T[which(N==0)],pch=18,xlab="M",ylab="Time (ns)",main="Binomial_Heap Assignment Opera
```



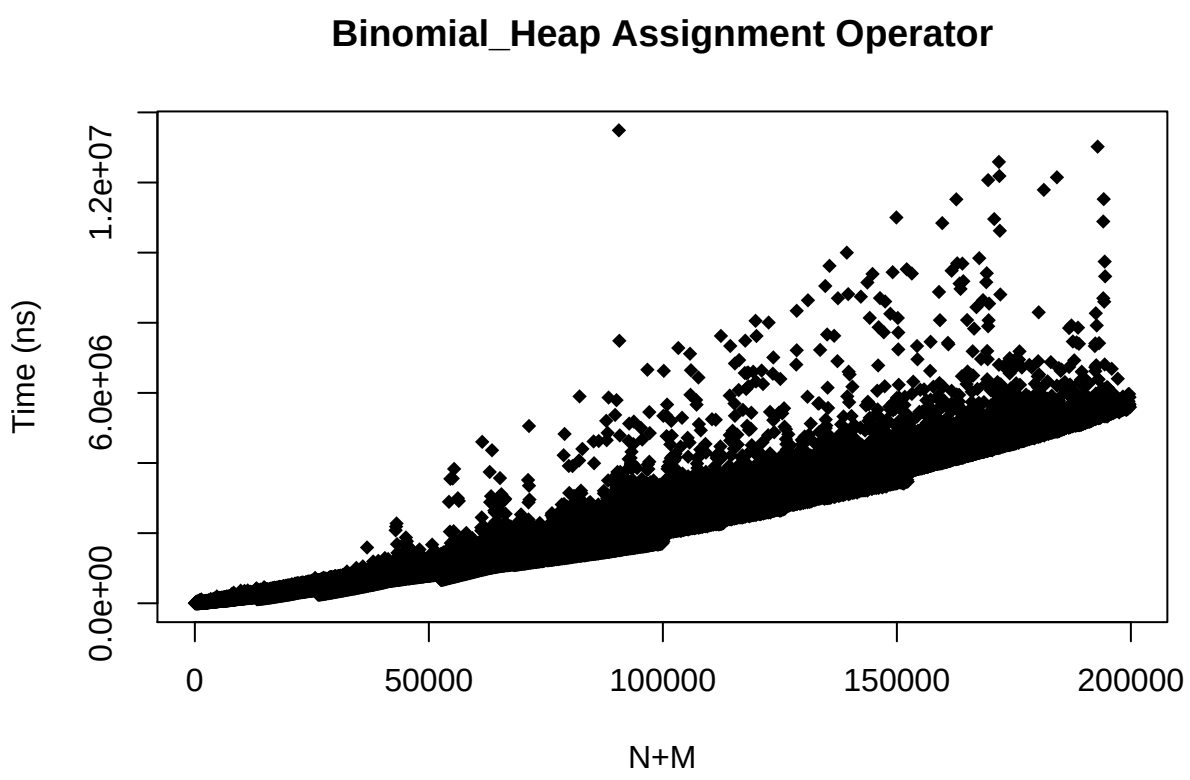

```
# This is extremely linear.  Wow!
plot(M[which(N==49900)],T[which(N==49900)],pch=18,xlab="M",ylab="Time (ns)",main="Binomial_Heap Assignme
```




```
plot(M[which(N==99900)],T[which(N==99900)],pch=18,xlab="M",ylab="Time (ns)",main="Binomial_Heap Assignme
```




```
# 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,T,pch=18,xlab="N+M",ylab="Time (ns)",main="Binomial_Heap Assignment Operator")
```




```
# We've already seen this.  Let's do multiple regression to take a look
# at the relationship between N+M and T using numbers.

cor(NplusM,T) # Higher correlation of 0.9891723.  A sign that we are probably seeing O(n+m), as we expe
```

```
## [1] 0.9891723
```

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

```
##
## Call:
## lm(formula = T ~ N + M)
##
## Coefficients:
## (Intercept)            N            M
##  -394063.94        25.74        30.18
```

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

```
##
## Call:
## lm(formula = T ~ NplusM)
##
## Coefficients:
## (Intercept)       NplusM
##  -394063.94        27.96
```

```
# Using our N and M variables seperately in multiple regression,
# we see that as N increases by 1, T increases by 25.74ns
```

```
# and as M increases by 1, T increases by 30.18ns.
# Using our N+M variable, we see that as N or M increases by 1,
# T increases by 27.96ns.
# The coefficient for our N+M variable is the average of the
# coefficients for each variable seperate.

# Because the coefficients are high and we have a strong correlation
# between N+M and T, we can assume that the assignment operator
# does in fact have a time complexity of O(n+m)

detach(assignment_operator_binomial)
```