

copy_ctor.R

Preston

2020-05-23

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

# the copy constructor should be should be  $O(n)$ 
# where  $n$  is the size of the array to be copied

copy_binomial = read.csv("./copy_ctor_binomial.csv")
attach(copy_binomial)

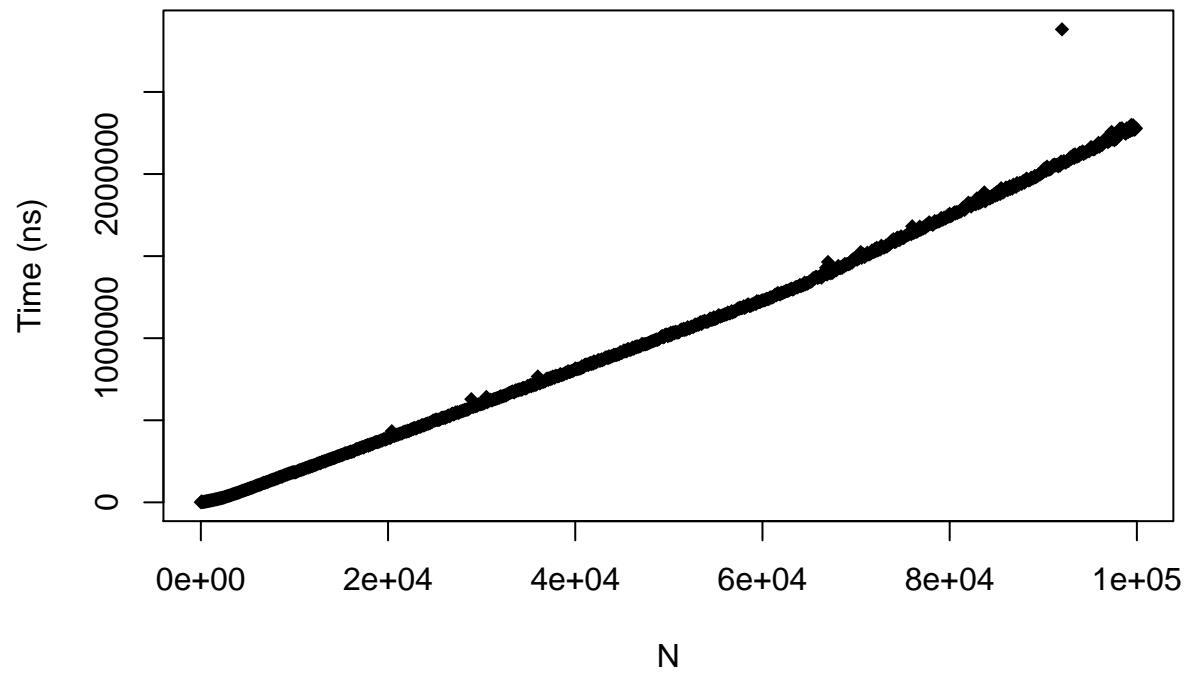
## The following object is masked from package:base:
##
##      T
summary(T)

##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      727   498590 1020990 1060626 1612424 2881307

# min 727
# q1 498590
# median 1020990
# mean 1060626
# q3 1612424
# max 2881307

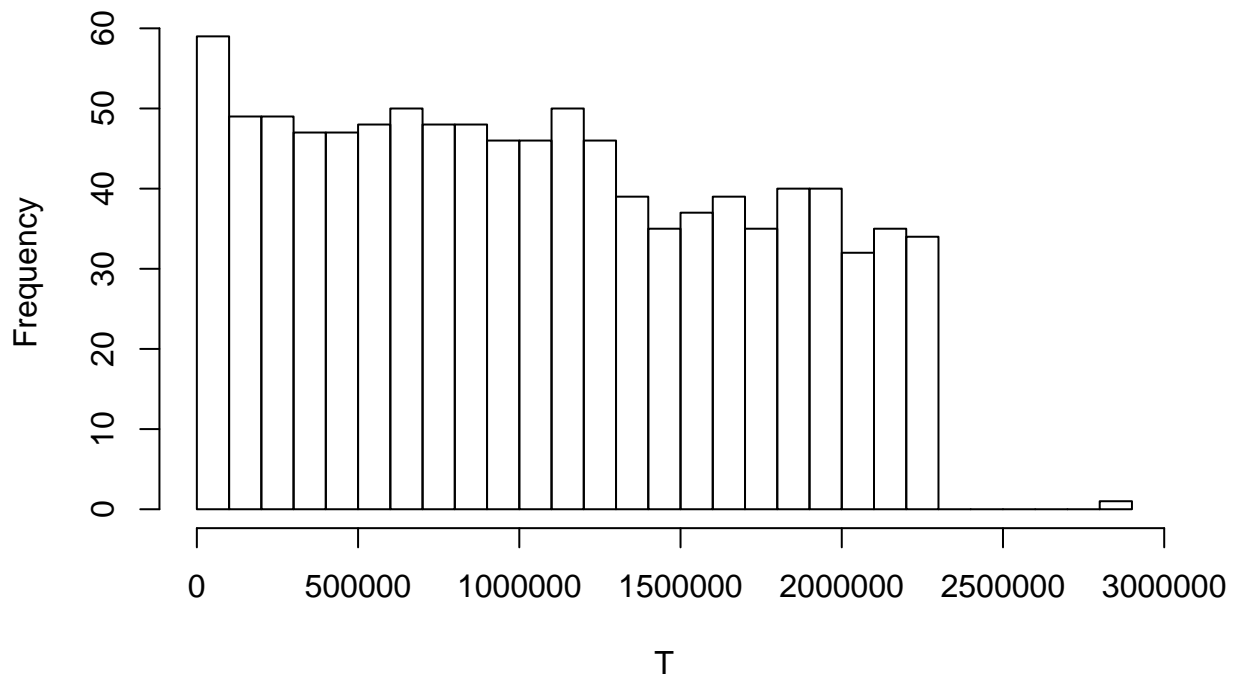
plot(N,T,pch=18,xlab="N",ylab="Time (ns)",main="Copy Constructor for Binomial Heap")
```

Copy Constructor for Binomial Heap



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

Histogram of T



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

```
sum(T>2500000) # There's only one point that seems to lie above the rest.
```

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

```
# this will not strongly affect our analysis
```

```
# Let's see if we can correlate N and T. It appears to be a linear relationship, as we expect.
```

```
cor(N,T) # very strong correlation of 0.9972907
```

```
## [1] 0.9972907
```

```
model = lm(T~N)
```

```
summary(model)
```

```
##
```

```
## Call:
```

```
## lm(formula = T ~ N)
```

```
##
```

```
## Residuals:
```

```
##      Min       1Q   Median       3Q      Max
## -66313 -34317  -2999   28319  863604
```

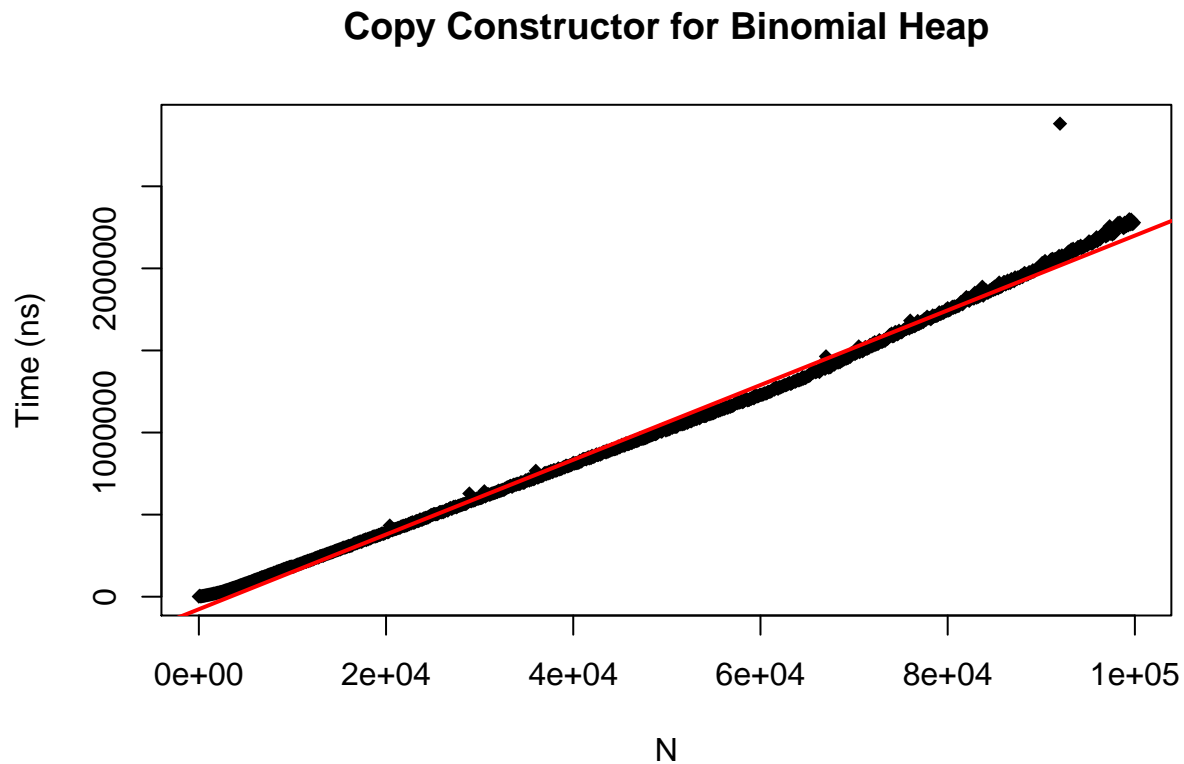
```
##
```

```
## Coefficients:
```

```
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -7.626e+04  3.066e+03  -24.87  <2e-16 ***
```

```
## N          2.276e+01  5.314e-02  428.29   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 48510 on 998 degrees of freedom
## Multiple R-squared:  0.9946, Adjusted R-squared:  0.9946
## F-statistic: 1.834e+05 on 1 and 998 DF,  p-value: < 2.2e-16
```

```
plot(N,T,pch=18,xlab="N",ylab="Time (ns)",main="Copy Constructor for Binomial Heap")
abline(model,lwd=2,col="red")
```



```
# The data seen for the copy constructor is clearly linear, which is expected given that
# to copy a heap, you must copy each element.
# Our linear regersion model tells us that for each additional element in the heap,
# copying takes about 22.7604 extra nanoseconds.
# Complexity is O(n)
```

```
detach(copy_binomial)
```