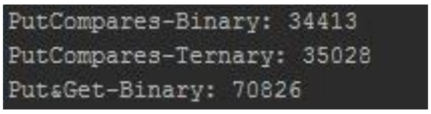
Ogj005

Compulsory assignment 2 – INF 102

Task1

- The task says that "...UTST.java only has a method put() for the purpose of this Experiment", so for this experiment I counted the numbers of compares in put() method for both tree structures. For UBST.java I also print a line that displays the total compares for both put() and get().
- For both structures the initial compare on key.CompareTo("another key") was counted in addition to all the if(cmp..) segments on that value.

Result for 1000words.txt:



```
PutCompares-Binary: 34413
PutCompares-Ternary: 35028
Put&Get-Binary: 70826
```

  - ○ Here we can see that the binary structure uses about 600 compares less than the ternary.
  - ○ Adding compare-count for get() in binary we can see that the amount of compares almost doubles. This was a random order search on every word in 1000words.txt, though order should not affect the binary search performance.

Task 2

Using the two approaches given in the text I compared the number of disks used when generating 200 random file sizes. The first approach with just inserting the files directly used 137 disks, compared to the sort first and then insert approach with 133 disks used. A slight improvement in storage usage. The difference increases with a greater number of files.

Task 3

Look at the included code for documentation.

Extra: I use the help classes BinaryIn and BinaryOut to write the binary code to file, in addition to this I use a "end of file" character to determine when the actual compressed data ends when decoding (throws away the 0 padding from writing bytes to file).