Coursework 1 for Theory 2 (104P)

To be submitted by 12.00 on Tuesday 1 March 2016

Please attempt all four questions. Marks will be deducted for answers that are not clear and concise. Briefly define any notation you use, but it is not necessary to add comprehensive explanatory text. Aim to give your answer in 4 pages (or less) of A4. Please staple your answers together, and please do NOT insert into a plastic folder.

Whilst it is not necessary, I recommend that you try to implement at least one of the algorithms you require using Python. If you do implement an algorithm, please include the Python code in your answer or email the Python code to me.

[**Question 1**] For the following piece of text, give a code trie formed using Huffman coding. Do ensure that the punctuation, upper and lower case letters, and spaces are considered. Also give the partially constructed trie that results from each cycle of the Huffman algorithm.

```
Why are what and when used?
```

[**Question 2**] Assuming we start with an empty B-tree tree of degree 1, give each intermediate B-tree, and the final B-tree, constructed from the following sequence of insertions of keys.

$$1, 2, 3, 4, 5, 6, 7, 8, 9$$

[**Question 3**] Assuming we start with an empty binary search tree, give each intermediate binary search tree, and the final binary search tree, all with the binary search tree property holding, formed from the following sequence of insertions of keys.

$$\texttt{Paris}, \texttt{Berlin}, \texttt{Rome}, \texttt{Oslo}, \texttt{Athens}, \texttt{Dublin}, \texttt{Helsinki}, \texttt{Luxembourg}, \texttt{Vienna}, \texttt{Warsaw}$$

[**Question 4**] Draw the binary tree representation of the following heap, and then apply the BuildHeap algorithm. For the output, assume that we want the key at each node to be greater than the key at its children nodes. Give each intermediate tree obtained by each call made to the Heapify algorithm.

$$1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15$$