Review Problems 4

1. Prove that the number of elements in the left and right sub-trees produced by the binary search (studied in lectures) are given by

$$L = \left\lfloor \frac{n-1}{2} \right\rfloor = \left\lceil \frac{n}{2} \right\rceil - 1 \tag{1}$$

$$R = \left\lceil \frac{n-1}{2} \right\rceil = \left\lfloor \frac{n}{2} \right\rfloor \tag{2}$$

2. Give an example of a tree that is not complete, still, with leaves only at the last two levels.

3. Some authors, e.g., (Knuth, 1997, Sec. 2.3.4.5, Eq. (4)), derive the internal path length as:

$$I_n = (n+1)\, q - 2^{q+1} + 2, \text{ where} \tag{3}$$

$$q = \lfloor \lg (n+1) \rfloor \tag{4}$$

   This is similar to what we have derived in lectures, except that we had $q = \lfloor \lg n \rfloor$.

   (a) Show that $\lfloor \lg (n+1) \rfloor$ and $\lfloor \lg n \rfloor$ are not always equal to each other. When are they equal?

   (b) Show that although $\lfloor \lg (n+1) \rfloor \neq \lfloor \lg n \rfloor$, both expressions for internal path are exactly the same.

4. Show that for a very large number of elements, the average case of of both the successful and unsuccessful Binary Search is approximately equal to $\lfloor \lg n \rfloor$.

5. If we measure the complexity of the Binary Search, studied in lectures, in terms of the step `mid = (last + first)/2`, does any thing change in our analyses?

6. Assume that we measure the complexity of the Binary Search, studied in lectures, in terms of the number of arithmetic comparisons, i.e., if we use $<$ and $==$ for the same key we count them as two not just one. Derive an expression for the complexity of successful and unsuccessful cases (best, worst, and average for each). After calculating the exact complexity, give the asymptotic complexity in terms of O-notation.

   Hint: If an element is not found at level l this means that extra l comparisons (in the previous levels) of type $<$ are counted.

7. Draw the tree produced by the sequential search.

8. Consider the `Binary1Search` version in the book.

   (a) Assume that we measure the complexity in terms of compared keys (as done in lectures). What is the complexity of this algorithm for all cases (successful (beast, worst, and average) and unsuccessful (best, worst, and average)).

   (b) Repeat the previous part assuming that we measure the complexity in terms of the number of arithmetic comparisons.

   (c) Express the complexity of both parts in terms of O notation.

   (d) Which version is better?

   Hint: Use the exact analysis as we did in lectures (not the approximations of the book).

9. Solve (Rosen, 2007, Prob. 3, Sec. 3.3), once by assuming key-comparison and another time by assuming arithmetic comparison. (Assume the use of `Binary2Search` of lectures).

References

Knuth, D. E. (1997), *The art of computer programming*, Reading, Mass.: Addison-Wesley, 3rd ed.

Rosen, K. H. (2007), *Discrete mathematics and its applications*, Boston: McGraw-Hill Higher Education, 6th ed.