Homework 4

RT Hatfield

14 September 2016

1. - $1 < \log_2 5$ therefore $5T(\lceil \frac{n}{2} \rceil) + O(n) = O(n^{\log_2 5}) \approx O(n^{0.431})$ This is my choice.
   - $0 < \log_{\frac{n}{n-1}} 2 = \frac{\ln 2}{\ln n - \ln(n-1)}$ therefore $2T(\lceil \frac{n}{\frac{n}{n-1}} \rceil) + O(1) = O(n^{\log_{\frac{n}{n-1}} 2})$ This one depends on the size of n. For $n = 100$, it evaluates to approximately $O(n^{68.97})$. It increases proportional to n.
   - $2 > \log_3 9$ therefore $9T(\lceil \frac{n}{3} \rceil) + O(n^2) = O(n^2)$
2. - $0 < \log_3 2$, $O(n^{\log_3 2})$
   - $1 < \log_4 5$, $O(n^{\log_4 5})$
   - $1 = \log_7 7$, $O(n \log n)$
   - $2 > \log_3 9$, $O(n^2)$
   - $3 = \log_2 8$, $O(n^3 \log n)$
3. If I understand this correctly, we want to join subproblems in constant time. Also, $0 = \log_b a$, so $a = 1$. So each recursion must involve a single subproblem of any size. Logical OR runs in constant time, and each recursion has a single subproblem.

   Data: A sorted list L and starting index i
   Result: Whether the list contains any item equal to its own position
   if $i < L.size()$ **then**
   | return $L[i] == i$ OR listInspector(L, $i + 1$);
   end
   else
   | return false;
   end