Tutorial 06 - Searching & Tries
CS 240E Winter 2022
University of Waterloo
Monday, February 14th, 2022

1. **Interpolation Search:**
This assignment will guide you towards a proof that a different modification of interpolation search also has expected run-time $O(\log \log n)$. Consider the modification shown in Algorithm 1 below, which compares not only at $A[m]$, but also at two indices m_ℓ and m_r that are roughly $\sqrt{N}$ indices to the left and right of m, and repeats in the appropriate sub-array.

Algorithm 1: *interpolation-search-3way*(A, n, k)

Input: Sorted array A of n integers, key k
if $(k < A[0])$ **then return** *"not found, would be left of index 0"*;
if $(k > A[n-1])$ **then return** *"not found, would be right of index $n-1$"*;
if $(k = A[n-1])$ **then return** *"found at index $n-1$"*;
$\ell \leftarrow 0,\ r \leftarrow n-1$;
while $r \geq \ell + 2$ **do** // inv: $A[\ell] \leq k < A[r]$
  $N \leftarrow r - \ell - 1, \quad p \leftarrow \frac{k - A[\ell])}{A[r]-A[\ell]}, \quad \mu \leftarrow p \cdot N,\ m \leftarrow \ell + \lceil \mu \rceil$;
  $m_\ell \leftarrow \max\{\ell, m - \lfloor \sqrt{N} \}$; $m_r \leftarrow \min\{r, m + \lfloor \sqrt{N} \rfloor\}$;
  if $(k < A[m_\ell])$ **then** $r \leftarrow m_\ell$;
  else if $(k < A[m]$ **then** $\ell \leftarrow m_\ell, r \leftarrow m$;
  else if $(k < A[m_r]$ **then** $\ell \leftarrow m, r \leftarrow m_r$;
  else $\ell \leftarrow m_r$;
end
// $r \leq \ell + 1$ and $A[\ell] \leq k < A[r]$, so $r = \ell + 1$ and k can only be $A[\ell]$
if $(k = A[\ell])$ **then return** *"found at index ℓ"*;
else return *"not found, would be between index ℓ and $\ell + 1$"*;

---

a) Assume that the items in A were randomly and uniformly chosen. Consider one execution of the while-loop, and call search-key k *good* $A[m_\ell] \leq k < A[m_r]$ and *bad* otherwise. Show that $P(k \text{ is good}) \geq \frac{3}{4}$.

You may assume that all items in A are distinct. You may also ignore rounding issues, i.e., assume N is a perfect-square and μ is an integer.

(Hint: Define $idx(k)$ and $offset(k)$ as we did in class and use the properties of $offset(k)$ that we derived there.)

b) Let $T(n)$ be the expected run-time on n items if items in A were randomly and uniformly chosen, Argue that $T(n)$ satisfies the recursion $T(n) \leq T(\sqrt{n}) + O(1)$.

You may make the same assumptions as in the previous part, and also use without proof that $T(n)$ is monotone, i.e., $T(n-1) \leq T(n)$.

Hint: What is the run-time if k is good? What if k is bad?

Note that the last part implies that $T(n) \in O(\log \log n)$ as shown in class.

2. **Numbers & Tries:**
Consider sorting the following base-4-numbers: 300, 211, 112, 230, 1, 0, 12, 101, 233, 110.

   a) Illustrate how you would sort them with MSD-radix sort, by drawing the recursion tree and the subarray in each recursion.

   b) Show the corresponding 4-way pruned trie.

   c) Show that the expected time to insert a base-4-number into a 4-way pruned trie is less than $\log_4 n + O(1)$, assuming all numbers have been uniformly chosen. You may assume the numbers have been padded with 0s so that all numbers begin with the same place value.

3. **Words & Tries:**
Suppose we have n English words (26-letter alphabet), where the combined length of all words is ℓ. Give an algorithm to sort the strings in $O(\ell)$ time in lexicographical ordering, e.g., "a" < "ab" < "b".