CSC263H Tutorial 5

Sample Solutions

Winter 2024

1. (a) Hash every name into a hash table of size roughly n. This already detects whether or not any two keys are identical. In the worst-case, this requires time $\Theta(i)$ for the i-th insertion, so total time $\Theta(n^2)$. But on average, we expect that each insertion will take time $\Theta(1)$, so the total time is only $\Theta(n)$. Given what we know of hashing – the worst-case is actually rare in practice – this is an excellent solution.

   (b) In this situation, a hash table would be no better than the original unsorted array: using it would require us to generate the list of sorted names using a method like selection sort (linear search for the first name, followed by linear search for the second name, etc.). This would take total time $\Theta(n^2)$ – worse than just sorting the original input list.

2. (a) **Linear Probing:**

key :	22	88			4	15	28	17	39	31	10
index :	0	1	2	3	4	5	6	7	8	9	10

   Quadratic Probing:

key :	22	88			4	15	28	17		31	10
index :	0	1	2	3	4	5	6	7	8	9	10

   * key 39 cannot be inserted: every probe location fails.

   Double Hashing:

key :	22		39	17	4	15	28	88		31	10
index :	0	1	2	3	4	5	6	7	8	9	10

   (b) See Section 11.4 in CLRS (or David's course notes on page 52).

   (c) See Section 11.4 in CLRS (or David's course notes on page 52).

   (d) See the discussion in Section 11.4 in CLRS (or David's course notes on page 53).