# Exercise Session 07

**Exercise 1.**
(CLRS 11.1-1) Suppose that a dynamic set $S$ is represented by a direct-address table $T$ of length $m$. Describe a procedure that finds the maximum element of $S$ . What is the worst-case performance of your procedure?

**Solution 1.**
Recall that direct-address tables work under the assumption that the set of actual keys is $K = \{0, \ldots, m-1\}$, that is, they represent dynamic sets containing elements in the range $0, .., m-1$. The maximum element can be found by performing a linear search starting from the slot $m-1$ down to 0 searching for the first occupied slot in the table. For the following algorithm we assume that the table represents a non-empty set, otherwise the procedure returns NIL.

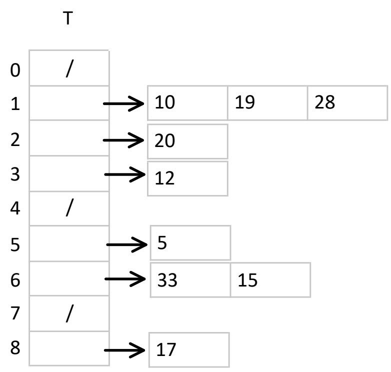
DIRECT-ADDRESS-MAXIMUM($T$)

1 $k = m - 1$
2 **while** $k \geq 1$ and DIRECT-ADDRESS-SEARCH($T, k$) = NIL
3 $\quad k = k - 1$
4 **return** DIRECT-ADDRESS-SEARCH($T, k$)

Note that the while loop terminates as soon as $k = 0$ or the $k$-th slot is not empty. Therefore, when whe execute line 4 we return either the first nonempty slot found stating from $m - 1$ or whatever element is present in the 0-th slot. The worst-case occurs when the maximum element in $T$ is 0 leading the algorithm to scan all the slots of $T$. Therefore the worst-case running time of DIRECT-ADDRESS-MAXIMUM($T$) is $O(m)$.

**Exercise 2.**
(CLRS 11.2-2) Demonstrate what happens when we insert the keys $5; 28; 19; 15; 20; 33; 12; 17; 10$ into a hash table with collisions resolved by chaining. Let the table have 9 slots, and let the hash function be $h(k) = k \mod 9$.

**Solution 2.**
After inserting the keys $5; 28; 19; 15; 20; 33; 12; 17; 10$ into an initially empty hash table with 9 slots, we end up in the following configuration




**Exercise 3.**
(CLRS 11.2-5) Suppose that we are storing a set of $n$ keys into a hash table of size $m$. Show that if the keys are drawn from a universe $U$ with $|U| > nm$, then $U$ has a subset of size $n$ consisting of keys that all hash to the same slot, so that the worst-case searching time for hashing with chaining is $\Theta(n)$.
*Hint:* The Dirichlet's box principle —a.k.a. pigeon hole principle— states that for $n, m \in \mathbb{N}$, if $nm+1$ objects are distributed among $m$ sets, then at least one of the sets will contain at least $n+1$ objects.

**Solution 3.**
In hash tables, the redistribution of objects in the slots is done by the hash function. In this context, the Dirichlet's box principle can be read as follows: regardless from the choice of the hash function $h$, there exist a subset of keys $K \subseteq U$ such that $|K| > n$ and a slot $0 \le j \le m$, such that for all $k \in K$, $h(k) = j$. The worst-case searching time occurs when all the $n$ elements in the hash table belong to $K$. In this case the list $L$ pointed by the slot $j$ has size $n$. We know that the worst-case searching time in a list of $n$ elements is $\Theta(n)$, therefore also the worst-case searching time in a hash table is $\Theta(n)$.

**Exercise 4.**
(CLRS 11.4-1) Consider inserting the keys 10; 22; 31; 4; 15; 28; 17; 88; 59 into a hash table of length $m = 11$ using *open addressing* with the auxiliary function $h'(k) = k$. Illustrate the result of inserting these keys using linear probing, using quadratic probing with $c_1 = 1$ and $c_2 = 3$, and using double hashing with $h_1(k) = k$ and $h_2(k) = 1 + (k \mod (m-1))$.

**Solution 4.**
Recall that the hash functions under linear probing, quadratic probing, and double hashing are

$$h(k,i) = (h'(k) + i) \mod m \qquad \text{(LINEAR PROBING)}$$
$$h(k,i) = (h'(k) + c_1 i + c_2 i^2) \mod m \qquad \text{(QUADRATIC PROBING)}$$
$$h(k,i) = (h_1(k) + i h_2(k)) \mod m \qquad \text{(DOUBLE HASHING)}$$

Let first consider inserting the keys 10; 22; 31; 4; 15; 28; 17; 88; 59 using linear probing. Key 10 goes in slot $h(10,0) = 10$ because it is free. Key 22 goes in slot $h(22,0) = 22 \mod 11 = 0$, key 31 goes in slot $h(31,0) = 9$, and key 4 goes in slot $h(4,0) = 4$, all because they find a free slot at the first probe. Key 15 tries to go in slot $h(15,0) = 15 \mod 11 = 4$ but it finds it occupied, then it tries the next probe $h(15,1) = (15+1) \mod 11 = 5$ and finds a free slots where to go. The insertions proceed in this way yielding the final table configuration $T = [22, 88, \text{NIL}, \text{NIL}, 4, 15, 28, 17, 59, 31, 10]$. Overall, the number of unsuccessful probes is 7. Below we show the step-by-step insertions.

| Step | Table configuration | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 00 | NIL | NIL | NIL | NIL | NIL | NIL | NIL | NIL | NIL | NIL | NIL |
| 01 | NIL | NIL | NIL | NIL | NIL | NIL | NIL | NIL | NIL | NIL | 10 |
| 02 | 22 | NIL | NIL | NIL | NIL | NIL | NIL | NIL | NIL | NIL | 10 |
| 03 | 22 | NIL | NIL | NIL | NIL | NIL | NIL | NIL | NIL | 31 | 10 |
| 04 | 22 | NIL | NIL | NIL | 4 | NIL | NIL | NIL | NIL | 31 | 10 |
| 05 | 22 | NIL | NIL | NIL | 4 | 15 | NIL | NIL | NIL | 31 | 10 |
| 06 | 22 | NIL | NIL | NIL | 4 | 15 | 28 | NIL | NIL | 31 | 10 |
| 07 | 22 | NIL | NIL | NIL | 4 | 15 | 28 | 17 | NIL | 31 | 10 |
| 08 | 22 | 88 | NIL | NIL | 4 | 15 | 28 | 17 | NIL | 31 | 10 |
| 09 | 22 | 88 | NIL | NIL | 4 | 15 | 28 | 17 | 59 | 31 | 10 |

Let now repeat the same insertion sequence using now quadratic probing. The final table configuration is $T = [22, \text{NIL}, 88, 17, 4, \text{NIL}, 28, 59, 15, 31, 10]$ and the total number of unsuccessful probes was 14. Below we show the step-by-step insertions

| Step | Table configuration | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 00 | NIL | NIL | NIL | NIL | NIL | NIL | NIL | NIL | NIL | NIL | NIL |
| 01 | NIL | NIL | NIL | NIL | NIL | NIL | NIL | NIL | NIL | NIL | 10 |
| 02 | 22 | NIL | NIL | NIL | NIL | NIL | NIL | NIL | NIL | NIL | 10 |
| 03 | 22 | NIL | NIL | NIL | NIL | NIL | NIL | NIL | NIL | 31 | 10 |
| 04 | 22 | NIL | NIL | NIL | 4 | NIL | NIL | NIL | NIL | 31 | 10 |
| 05 | 22 | NIL | NIL | NIL | 4 | NIL | NIL | NIL | 15 | 31 | 10 |
| 06 | 22 | NIL | NIL | NIL | 4 | NIL | 28 | NIL | 15 | 31 | 10 |
| 07 | 22 | NIL | NIL | 17 | 4 | NIL | 28 | NIL | 15 | 31 | 10 |
| 08 | 22 | NIL | 88 | 17 | 4 | NIL | 28 | NIL | 15 | 31 | 10 |
| 09 | 22 | NIL | 88 | 17 | 4 | NIL | 28 | 59 | 15 | 31 | 10 |

Using double hashing, the final table configuration is $T = [22, \textsc{Nil}, 59, 17, 4, 15, 28, 88, \textsc{Nil}, 31, 10]$, and the total number of unsuccessful probes was 7. Below we show the step-by-step insertions

| Step | Table configuration | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 00 | NIL | NIL | NIL | NIL | NIL | NIL | NIL | NIL | NIL | NIL | NIL |
| 01 | NIL | NIL | NIL | NIL | NIL | NIL | NIL | NIL | NIL | NIL | 10 |
| 02 | 22 | NIL | NIL | NIL | NIL | NIL | NIL | NIL | NIL | NIL | 10 |
| 03 | 22 | NIL | NIL | NIL | NIL | NIL | NIL | NIL | NIL | 31 | 10 |
| 04 | 22 | NIL | NIL | NIL | 4 | NIL | NIL | NIL | NIL | 31 | 10 |
| 05 | 22 | NIL | NIL | NIL | 4 | 15 | NIL | NIL | NIL | 31 | 10 |
| 06 | 22 | NIL | NIL | NIL | 4 | 15 | 28 | NIL | NIL | 31 | 10 |
| 07 | 22 | NIL | NIL | 17 | 4 | 15 | 28 | NIL | NIL | 31 | 10 |
| 08 | 22 | NIL | NIL | 17 | 4 | 15 | 28 | 88 | NIL | 31 | 10 |
| 09 | 22 | NIL | 59 | 17 | 4 | 15 | 28 | 88 | NIL | 31 | 10 |

**Exercise 5.**
(CLRS 11.4-3) Consider an open-address hash table with uniform hashing. Give upper bounds on the expected number of probes in an unsuccessful search and on the expected number of probes in a successful search when the load factor is 3/4 and when it is 7/8.

**Solution 5.**
By Theorem 11.6 in CLRS, the expected number of probes in an unsuccessful search is bounded by $1/(1-\alpha)$. Thus, for $\alpha = 3/4$ we have $1/(1-\alpha) = 4$, and for $\alpha = 7/8$ we have $1/(1-\alpha) = 8$.
By Theorem 11.8 in CLRS, the expected number of probes in a successful search is bounded by $(1/\alpha)\ln(1/1-\alpha)$. Hence, for $\alpha = 3/4$ we have $(1/\alpha)\ln(1/1-\alpha) = (4/3)\ln 4 \cong 1.85$, and for $\alpha = 7/8$ we have $(1/\alpha)\ln(1/1-\alpha) = (8/7)\ln 8 \cong 2.38$.