Algorithms: Homework 2

Li-Yuan Wei

May 4, 2023

Problem 1

Using Figure 8.3 as a model, illustrate the operation of RADIX-SORT on the following list of English words: NOD, HOG, SHY, BAN, BAR, JET, EBB, PAR, ASH, PET, TED, ROT, FIG.

Solution.

0(*initial*)	*digit* 1	*digit* 2	*digit* 3
NOD	EB**B**	B**A**N	**A**SH
HOG	NO**D**	B**A**R	**B**AN
SHY	TE**D**	P**A**R	**B**AR
BAN	HO**G**	E**B**B	**E**BB
BAR	FI**G**	T**E**D	**F**IG
JET	AS**H**	J**E**T	**H**OG
EBB	BA**N**	P**E**T	**J**ET
PAR	BA**R**	S**H**Y	**N**OD
ASH	PA**R**	F**I**G	**P**AR
PET	JE**T**	N**O**D	**P**ET
TED	PE**T**	H**O**G	**R**OT
ROT	RO**T**	R**O**T	**S**HY
FIG	SH**Y**	A**S**H	**T**ED

■

Problem 2

Show how to sort n integers in the range 0 to $n^2 - 1$ in $\mathcal{O}(n)$ time.

Solution. ■

Problem 3

Using Figure 8.4 as a model, illustrate the operation of BUCKET-SORT on the array A=<.49, .30, .56, .74, .68, .50, .69, .37, .41, .72>

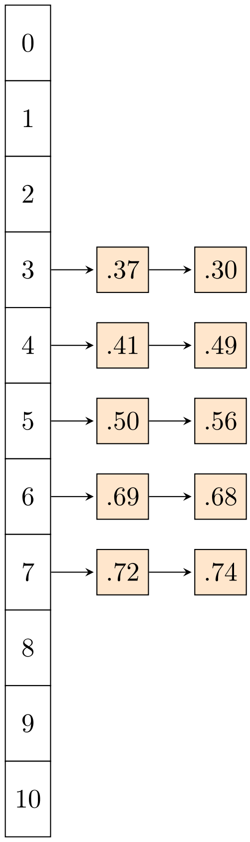

Figure 1: Insert data

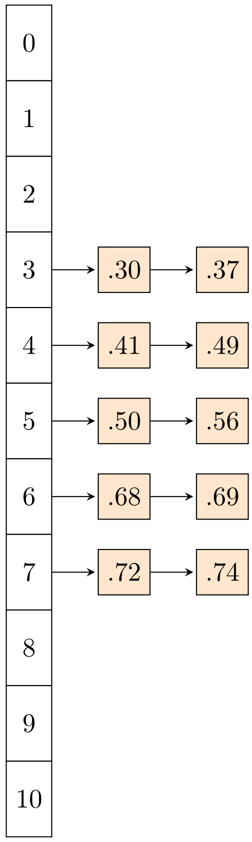

Figure 2: Sort each bucket

Solution. To get sorted data, chain all in Figure 2 from bucket 0 to bucket 10:
$< .30, .37, .41, .49, .50, .56, .68, .69, .72, .74 >$ ■

Problem 4

Suppose that you have a "black-box" worst-case linear time median subroutine. Give a simple, linear-time algorithm that solves the selection problem for an arbitrary order statistic.

Solution. Psuedocode for finding the k-th smallest element:

FIND-K-TH-SMALLEST(X, k)

if $i > A.heap\text{-}size$
  error $A[i]$ does not exist
key $= A[i]$
$A[i] = A[A.heap\text{-}size]$
$A.heap\text{-}size = A.heap\text{-}size - 1$
if $A[i] <$ key
  MAX-HEAPIFY(A, i)
else
  key1 $= A[i]$
  $A[i] =$ key
  HEAP-INCREASE-KEY$(A, i, key1)$

■

Problem 5

Let $X[1..n]$ and $Y[1..n]$ be two arrays, each containing n numbers already in sorted order. Give an $\mathcal{O}(\lg n)$-time algorithm to find the median of all $2n$ elements in arrays X and Y.

Solution. Psuedocode for finding the median of arrays $X[1..n]$ and $Y[1..n]$, xi is X array's starting index and xj is X array's last index; yi is Y array's starting index and yj is Y array's last index:

```
FIND-MEDIAN(X, Y, xi, xj, yi, yj, n)
if n == 1
    // Both arrays have only one element, pick the smaller one as median
    return MIN(X[1], Y[1])
xmid = X[(xi + xj)/2] // X's median
ymid = Y[(yi + yj)/2] // Y's median
if xmid < ymid
    // median must be greater than xmid and less than ymid
    return FIND-MEDIAN(X, Y, xmid + 1, xj, yi, ymid, n/2)
else
    // median must be less than xmid and greater than ymid
    return FIND-MEDIAN(X, Y, xi, xmid, ymid + 1, yj, n/2)
```

This algorithm reduce the array size by half every recursion call, similar with binary search. Thus, it's running time is $\mathcal{O}(\lg n)$. ■

Problem 6

Demonstrate the insertion of the keys 23, 38, 19, 40, 29, 43, 12, 27, 11 into a hash tale with collisions resolved by chaining. Let the table have 9 slots, and let the hash function be $h(k) = k \pmod 9$.

Solution. ■

(1) insert $23 \to 23 \pmod 9 = 5$

(2) insert $38 \to 38 \pmod 9 = 2$

(3) insert $19 \to 19 \pmod 9 = 1$

(4) insert $40 \to 40 \pmod 9 = 4$

(5) insert $29 \to 29 \pmod 9 = 2$, insert head

(6) insert $43 \to 43 \pmod 9 = 7$

(7) insert $12 \to 12 \pmod 9 = 3$

(8) insert $27 \to 27 \pmod 9 = 0$

(9) insert $11 \to 11 \pmod 9 = 2$, insert head

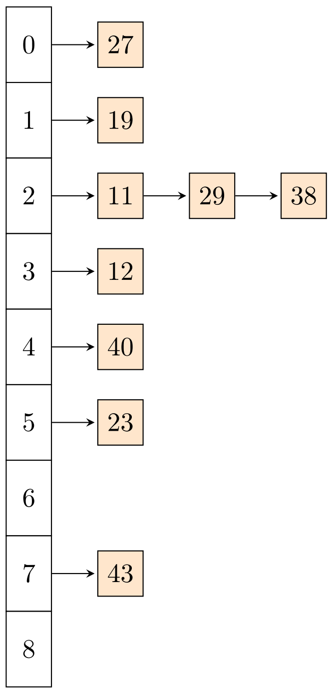

Figure 3: Hash table with chaining

Problem 7

Consider inserting the keys 10, 23, 31, 4, 12, 28, 17, 87, 58 into a hash table of length $m = 11$ using open addressing with the auxiliary hash function $h'(k) = k \pmod{m}$. Illustrate the result of inserting these keys using **linear probing**, using **quadratic probing** with $c1 = 1$, and $c2 = 3$, and using **double hashing** with $h_2(k) = 1 + (k \pmod{(m-1)})$.

(1) insert 10 $\rightarrow 10 \pmod{11} = 10$
(2) insert 23 $\rightarrow 23 \pmod{11} = 1$
(3) insert 31 $\rightarrow 31 \pmod{11} = 9$
(4) insert 4 $\rightarrow 4 \pmod{11} = 4$
(5) insert 12 $\rightarrow 12 \pmod{11} = 1$ $\times$ collision
insert 12 to slot 2 because slot 1 is occupied
(6) insert 28 $\rightarrow 28 \pmod{11} = 6$
(7) insert 17 $\rightarrow 17 \pmod{11} = 6$ $\times$
insert 17 to index 7 because slot 6 is occupied
(8) insert 87 $\rightarrow 87 \pmod{11} = 10$ $\times$
insert 87 to index 0 because slot 10 is occupied
(9) insert 58 $\rightarrow 58 \pmod{11} = 3$

0	87
1	23
2	12
3	58
4	4
5	0
6	28
7	17
8	0
9	31
10	10

Figure 4: linear probing

---

resolve collision: $h(k) + 1 * i + 3 * i^2 \pmod{11}$
(1) insert 10 $\rightarrow 10 \pmod{11} = 10$
(2) insert 23 $\rightarrow 23 \pmod{11} = 1$
(3) insert 31 $\rightarrow 31 \pmod{11} = 9$
(4) insert 4 $\rightarrow 4 \pmod{11} = 4$
(5) insert 12 $\rightarrow 12 \pmod{11} = 1$ $\times$ collision
$(1 + 12 + 3 * 144) \pmod{11} = (1 + 1 + 3) \pmod{11} = 5$
insert 12 to slot 5
(6) insert 28 $\rightarrow 28 \pmod{11} = 6$
(7) insert 17 $\rightarrow 17 \pmod{11} = 6$ $\times$
$(6 + 17 + 3 * 289) \pmod{11} = (6 + 6 + 9) \pmod{11} = 10$ $\times$
$(10 + 17 + 3 * 289) \pmod{11} = (10 + 6 + 9) \pmod{11} = 3$
insert 17 to slot 3
(8) insert 87 $\rightarrow 87 \pmod{11} = 10$ $\times$
$(10 + 87 + 3 * 87 * 87) \pmod{11} = (10 + 10 + 3) \pmod{11} = 1$ $\times$
$(1 + 87 + 3 * 87 * 87) \pmod{11} = (1 + 10 + 3) \pmod{11} = 3$ $\times$
$(3 + 87 + 3 * 87 * 87) \pmod{11} = (3 + 10 + 3) \pmod{11} = 5$ $\times$
$(5 + 87 + 3 * 87 * 87) \pmod{11} = (5 + 10 + 3) \pmod{11} = 7$
insert 87 to slot 7
(9) insert 58 $\rightarrow 58 \pmod{11} = 3$ $\times$
$(7 + 58 + 3 * 58 * 58) \pmod{11} = (7 + 3 + 5) \pmod{11} = 4$ $\times$
$(4 + 58 + 3 * 58 * 58) \pmod{11} = (4 + 3 + 5) \pmod{11} = 1$ $\times$
$(1 + 58 + 3 * 58 * 58) \pmod{11} = (1 + 3 + 5) \pmod{11} = 9$ $\times$
$(9 + 58 + 3 * 58 * 58) \pmod{11} = (9 + 3 + 5) \pmod{11} = 6$ $\times$
$(6 + 58 + 3 * 58 * 58) \pmod{11} = (6 + 3 + 5) \pmod{11} = 3$ $\times$
$(3 + 58 + 3 * 58 * 58) \pmod{11} = (3 + 3 + 5) \pmod{11} = 0$
insert 58 to slot 0

0	58
1	23
2	0
3	17
4	4
5	12
6	28
7	87
8	0
9	31
10	10

Figure 5: quadratic probing

Problem 8

Solve the following assembly-line problem:
$e_1 = 1, e_2 = 1, x_1 = 7, x_2 = 3$
$a_{1,1} = 3, a_{1,2} = 2, a_{1,3} = 5, a_{1,4} = 4, a_{1,5} = 2$
$a_{2,1} = 4, a_{2,2} = 2, a_{2,3} = 4, a_{2,4} = 6, a_{2,5} = 4$
$t_{1,1} = 1 t_{1,2} = 2, t_{1,3} = 3, t_{1,4} = 1$
$t_{2,1} = 2, t_{2,2} = 3, t_{2,3} = 1, t_{2,4} = 2.$

Solution. ■

Problem 9

Find an optimal parenthesization of a matrix-chain product whose sequence of dimension is <5, 7, 3, 2, 4, 5>

Solution. ■

Problem 10

Determine an LCS of <1, 1, 0, 1, 0, 1> and <0, 0, 1, 1, 0, 1, 1>.

Solution. ■