CSCE 500 Midterm Exam #2 10/25/21

1. Given that for an open-address hash table with load factor $\alpha = n/m < 1$, the expected number of probes in <u>unsuccessful search</u> under uniform hashing is at most $1/(1-\alpha)$, prove the expected number of probes in a <u>successful probe</u> under uniform hashing being at most $(1/\alpha)\cdot\ln(1-\alpha)^{-1}$ by giving a proof sketch which explains how many probes are needed to locate existing keys. (15%)

The number of successful inserts for the <u>second key</u> takes ≤ **1/(1-1/m)** probes.

The number of successful inserts for the <u>third key</u> takes ≤ **1/(1-2/m)** probes.

The number of successful inserts for the <u>fourth key</u> takes ≤ **1/(1-3/m)** probes.

The number of successful inserts for the <u>(i + 1)th key</u> takes ≤ **1/(1-i/m)**.

The expected number of successful inserts takes $\leq \frac{1}{n}\sum_{i=0}^{n-1}\frac{1}{1-\frac{i}{m}}$

$$\frac{1}{n}\sum_{i=0}^{n-1}\frac{1}{1-\frac{i}{m}} = \frac{1}{m\alpha}\sum_{i=0}^{n-1}\frac{m}{m-i} = \frac{1}{\alpha}\sum_{i=0}^{n-1}\frac{1}{m-i} \leq \frac{1}{\alpha}\int_{0}^{n-1}\frac{1}{m-x}dx$$

$$\frac{1}{\alpha}[\ln(m-0) - \ln(m-n+1] = \frac{1}{\alpha}\ln\frac{m}{m-n+1} =$$

Let m-n >> 1

$$\frac{1}{\alpha}\ln\frac{m}{m-n} = \frac{1}{\alpha}\ln\frac{1}{1-n/m} = \frac{1}{\alpha}\ln\frac{1}{1-\alpha}$$

2. Use <u>perfect hashing</u> to store the set of k = {10, 40, 64, 91}, with its outer hash function of $h(k)=((a\cdot k+b) \bmod p) \bmod m$, where a = 3, b = 45, p = 137, and m (i.e., the outer hash table size) = 8. Illustrate the perfect hashing result under k after devising the <u>appropriate inner hash function(s)</u> as needed. (15%)

$h_{3,45}(10) = [(3\cdot10+45) \bmod 137) \bmod 8] = 3$

$h_{3,45}(40) = [(3\cdot40+45) \bmod 137) \bmod 8] = 4$

$h_{3,45}(64) = [(3\cdot64+45) \bmod 137) \bmod 8] = 4$

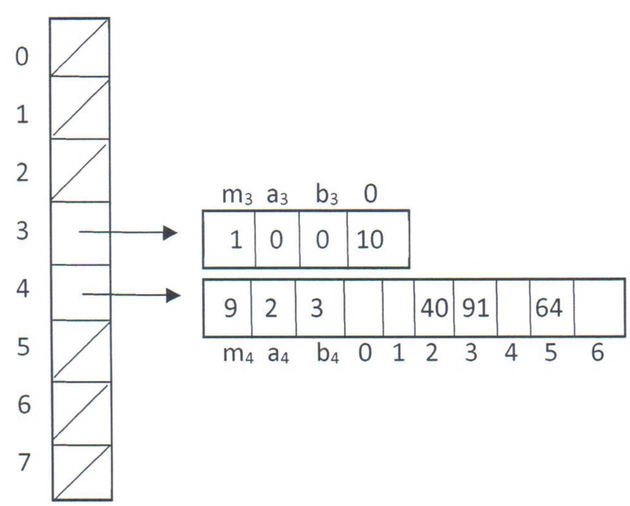
$h_{3,45}(91) = [(3\cdot91+45) \bmod 137) \bmod 8] = 4$

$h_{0,0}(10) = [(0\cdot10+0) \bmod 137) \bmod 8] = 0$

$h_{2,3}(40) = [(2\cdot40+3) \bmod 137) \bmod 8] = 2$

$h_{2,3}(64) = [(2\cdot64+3) \bmod 137) \bmod 8] = 5$

$h_{2,3}(91) = [(2\cdot91+3) \bmod 137) \bmod 8] = 3$



3. (a) Explain briefly how Cuckoo hashing works under two hash functions of h_1 and h_2. (10%)

Cuckoo hashing uses two tables, T_1 and T_2, with two unique hash functions h_1 and h_2. Initially, keys are hashed into table T_1 using hash function h_1. However, if a subsequent key is hashed into the same location as previously hashed, it will be removed , replaced with the subsequent key and hashed into table T_2 using hash function h_2. If there is a collision is table T_2, the current key in the location will be removed and replaced with the new key. The subsequent key will be hashed into table T_1 using hash function h_1.

(b) State the situation when a new key cannot be inserted in a Cuckoo hash table successfully; provide two solutions for key insertion failures and contrast them in terms of advantages/disadvantages. (8%)

It is possible that a key may not be able to be inserted into either table if the new edge (defined by the new key) contains at most one cycle.

Solution 1: Increase the table sizes.

- Advantage: Decreases the chances of key insertion failures.
- Disadvantages: Takes up more memory.

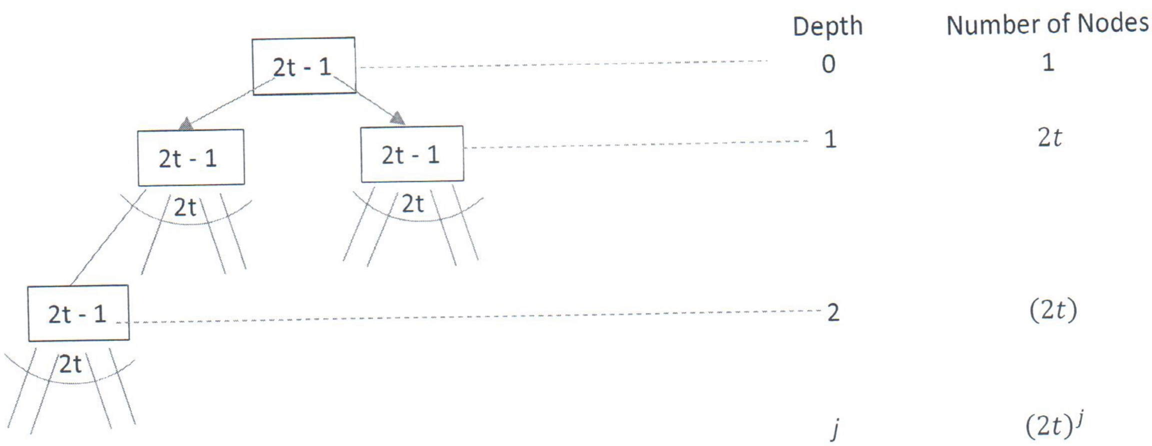
Solution 2: Use quadratic hashing.

- Advantage: Separates the spacing between keys, making collisions less likely.
- Disadvantages: The hashing becomes more complicated and still does not guarantee that insertions will be possible.

4. Deletion in a binary search tree relies on TRANSPLANT procedure given below, where the subtree rooted at u is replaced by the subtree rooted at v. Complete the three missing statements of the procedure and provide an illustrative figure to show the resulting figure after the procedure is conducted. (12%)

```
TRANSPLANT(T, u, v)
if u.p == NIL
    T.root = v
elseif u == u.p.left
    u.p.left = v
else u.p.right = v
if v ≠ NIL
    v.p = u.p
```

5. For a B-tree of height h with the minimum node degree of t ≥ 2, derive the maximum number of keys that can be stored in such a B-tree (10%)



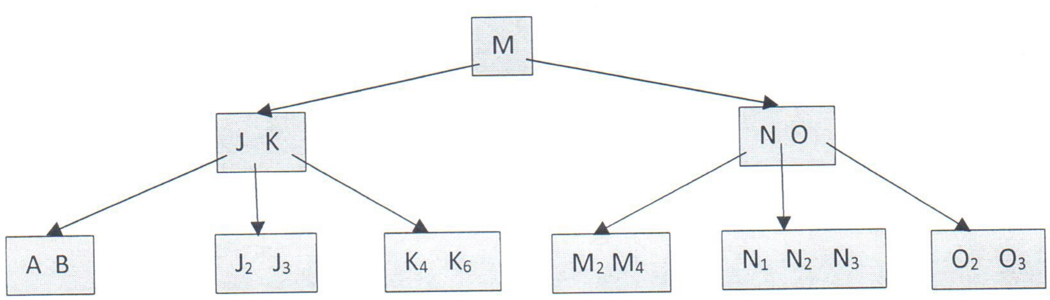
The maximum number of keys: $n \leq (keys\ per\ node)(number\ of\ nodes)$

$$n \leq (2t-1)\sum_{i=1}^{h}(2t)^i$$

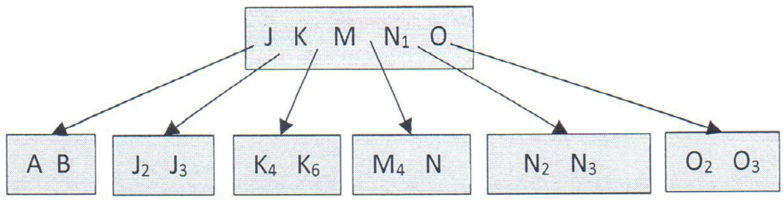
$$n \leq (2t-1)\frac{1-(2t)^{h+1}}{1-2t}$$

$$n \leq (2t)^{h+1} - 1$$

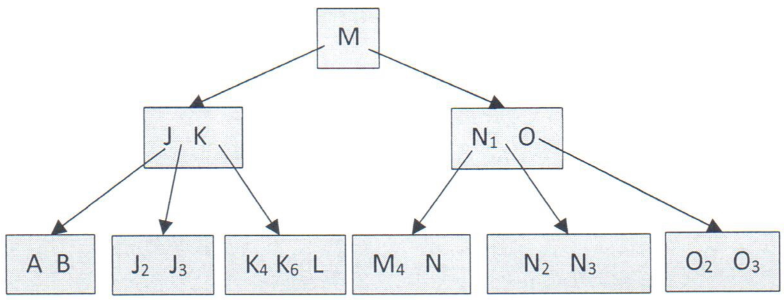
6. Given the initial B-tree with the minimum node degree of <u>t = 3</u> below, show the results (a) <u>after deleting</u> the key of M_2, (b) followed by inserting the key of L, (c) then by deleting the key of J_2, (d) then by inserting the key of O1, with $O < O_1 < O_2$, (e) then deleting K, and (f) then by deleting M. (Show the result after each deletion and after each insertion. 18%)



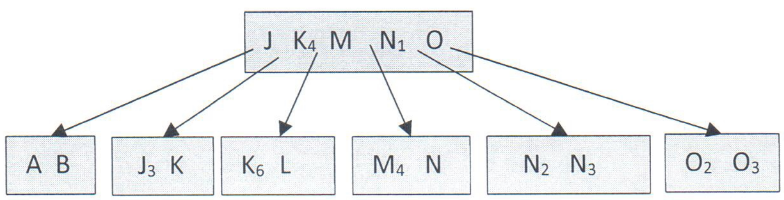
Deleting M_2...



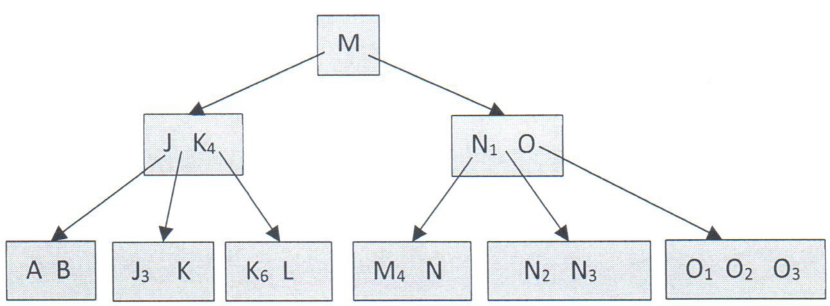
Inserting L...



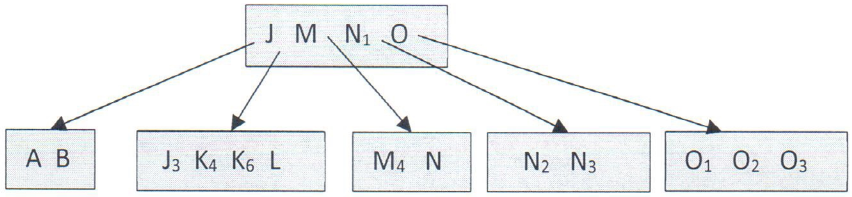
Deleting J_2...



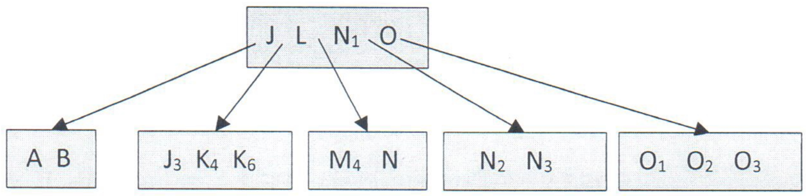
Inserting O_1...



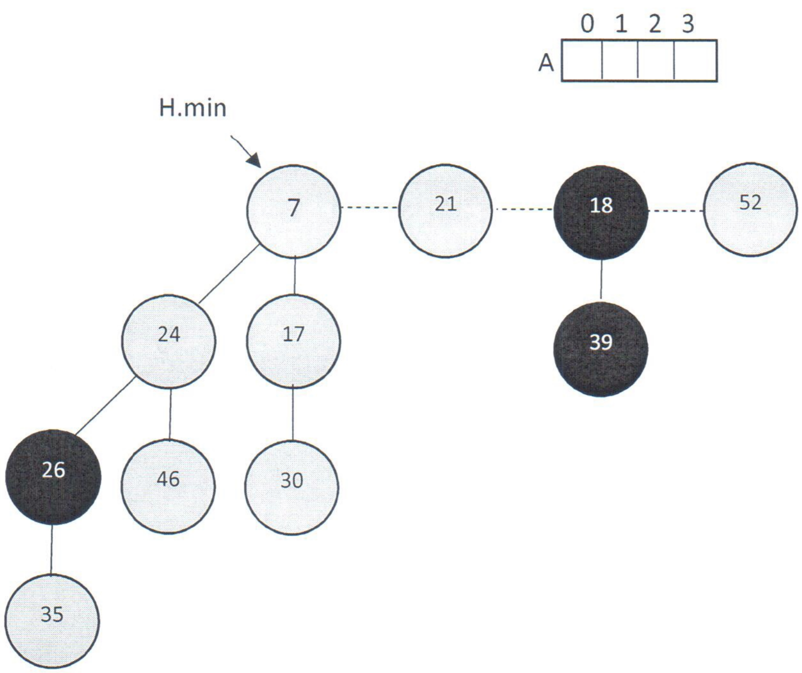
Deleting K…

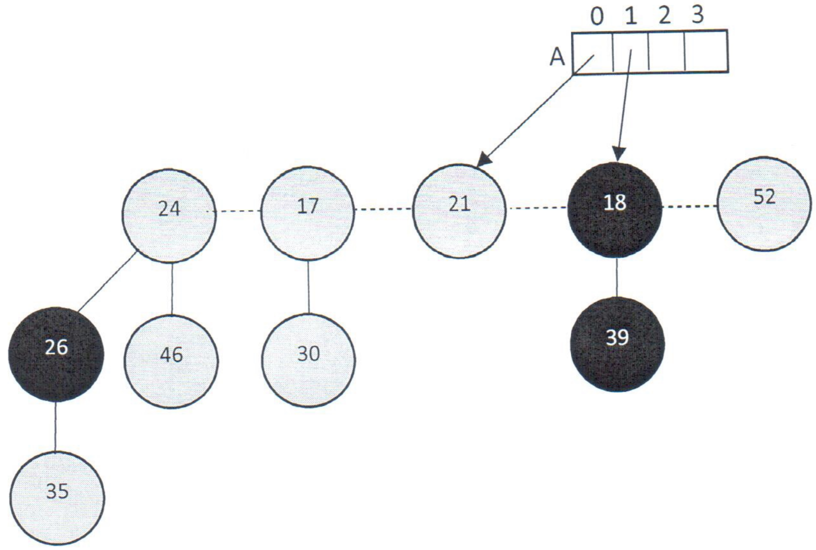


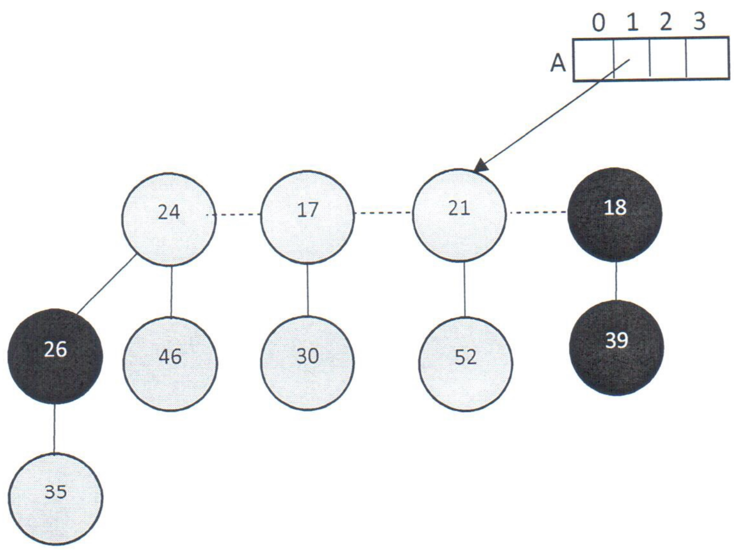
Deleting M…

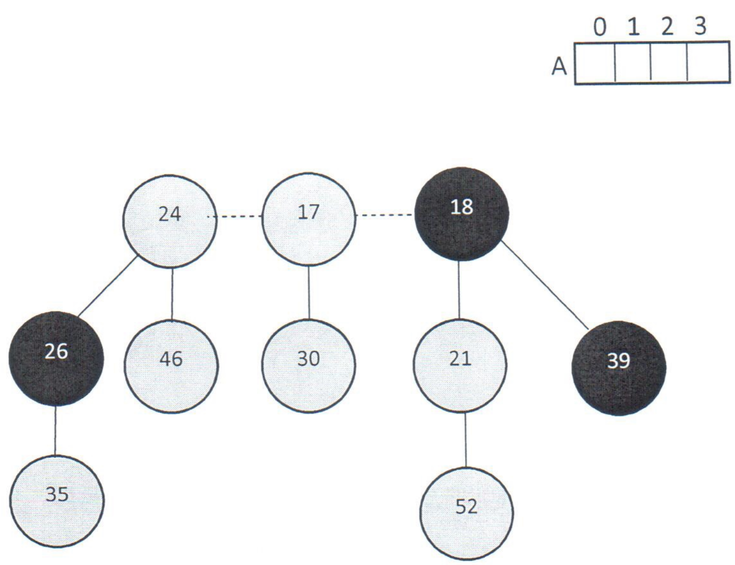


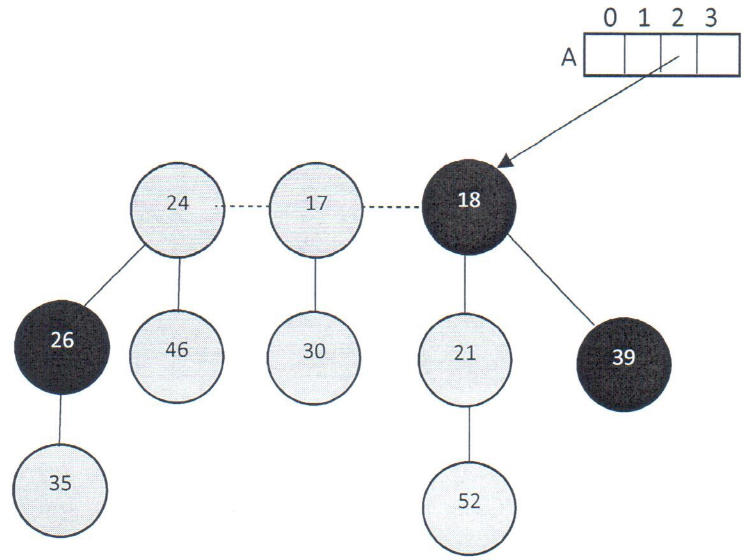
7. A Fibonacci min-heap relies on the procedure of CONSOLIDATE to merge min-heaps in the root list upon the operation of extracting the minimum node. Given the following Fibonacci min-heap, show every consolidation step and the final heap result after H.min is extracted with the aid of A (12%).

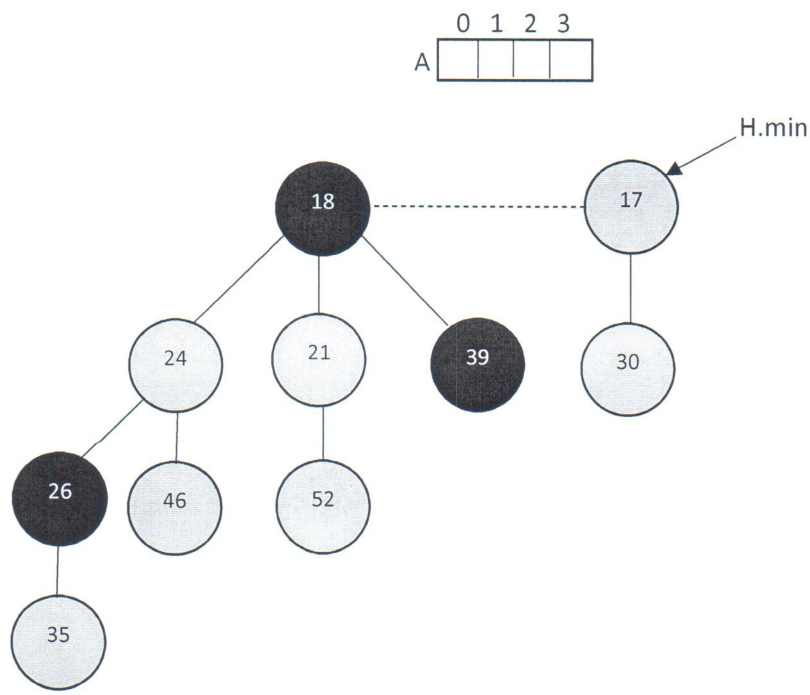



0 1 2 3
A
24
17
21
18
52
26
46
30
39
35


0 1 2 3
A
24
17
21
18
26
46
30
52
39
35


0 1 2 3
A
24
17
18
26
46
30
21
39
35
52


0 1 2 3
A
24
17
18
26
46
30
21
39
35
52


0 1 2 3
A
H.min
18
17
24
21
39
30
26
46
52
35