1. (12) State and explain clearly the four assumptions of the RAM computational model.

7

1) each operation has a constant time, such as assignment, comparsion!

2) each cell can only store a simple object!

3) each operation goes acess into cell ~~it's also~~, using a constant time!

4) each algorithm is ~~not~~ simple to see?!

Random Access

1. Only one instruction can be ~~exe~~ executed at a time,

2. Each datum is small enough to be stored in a ~~sig~~ single memory cell

3. Each stored datum can be accessed with the same constant cost

4. Each basic operations such as read, write, requires a constant cost!

2. (10) When implementing an ADT for a set of records S, $|S| = 2^6$, it is determined that a find operation, find(x,S), will requires 0.2 ms to execute. If the complexity of the find operation is given by the following closed-form expressions T(n), compute the time required to execute this operation when $|S| = 2^{12}$. *Remark*: You must simplify your answer for credit.

(a) $T(n) = 100^{1024}$.

9

$$\frac{T(n)}{C(n)} = \frac{T^*(n)}{C^*(n)}$$

$$\frac{100^{1024}}{0.2\times10^{-3}s} = \frac{100^{1024}}{C^*(n)}$$

$$C^*(n) = 0.2\,ms$$

(b) $T(n) = 560\lg n$.

$$\frac{560\lg 2^6}{0.2\,ms} = \frac{560\lg 2^{12}}{C^*(n)}$$

$$C^*(n) = \frac{\cancel{560}\times\cancel{12}\ 2\cancel{4}}{0.2\times\cancel{560}\times\cancel{6}}$$

(-1)

$$= 10\ \cancel{ms}$$

$$= 2\times0.2 = 0.4\,ms$$

$\cancel{0.2|2}$ $\frac{20}{2}$

1024 2048 4096

3. (16) If a set of 4090 records is being stored/organized using a binary tree T with 4090 nodes (one record per node), answer the following questions with ***integer solution*** if possible.

(a) What is the min height of T? What kind of tree structure of T will have a minimum height?

min height $= \lfloor \log_2 4090 \rfloor = 11$

completed binary tree!

15

(b) What is the maximum number of leaf nodes in T? What kind of tree structure of T will have a maximum number of leaf nodes?

max leaf nodes $= \lceil \frac{4090}{2} \rceil = 2045$

completed binary tree!

4095

Height h=11 $2^{12}-1$ nodes

2^{11} leaves 2048

remove 5 leaves, add two leaves

2048 − 5 + 2 = 2045 leaves

(c) If T is being implemented using the sequential array implementation as discussed in class, what is the size of array A in order to store T? What kind of tree structure of T will require an array of maximum size to store it?

min: size: $2^{12}-1$ min height = 11

Right skew ~~tree!~~ −1 max height = 4089

~~maximum size~~ R skew tree : max size: $2^{4090}-1$

(d) In the extended binary tree T_E of T, what is the minimum and maximum number of external nodes in T_E?

minimum = maximum = 4091

4. (12) Answer the following questions.

(a) What are the two most important characteristics of a "good" hash function h(x)?

10

compute time is constant $\Theta(1)$

it will distribute n objects evenly in m table size.

(b) What are the two biggest problems when using closed hash table?

primary cluster / secondary cluster → insertion and deletion is hard!

−2

~~how to choose good table size.~~

search is hard, ~~when deleting a value~~

(c) Recall that in monitoring the performance of a hash table, whenever λ approaches a constant value k, rehashing should be performed.

(i) What is k in an open hash table?

1

(ii) What is k in a closed hash table?

$\frac{1}{2}$

① Insertion may fail even table is not full, waste memory

② Searching needs to be continued even when empty (by deletion) bucket encountered: inefficient

③ May form secondary cluster during insertion resulting in increasing search time

5. (8) Using the hash function $h(x) = x \bmod m$ and quadratic probing to construct a (closed) hash table H with $m = 11$ buckets by inserting a set of 8 records with keys {18, 35, 25, 4, 22, 13, 36, 24}, in the given order, into H. *Remark:* You must show your computations for locations and illustrate the final hash table H clearly for credit.

$\boxed{6}$ ✓

Address computations:

① $18 \% 11 = 7$

② $35 \% 11 = 2$

③ $25 \% 11 = 3$

④ $4 \% 11 = 4$

⑤ $22 \% 11 = 0$

⑥ $13 \% 11 = 2$ collision.

~~$(2+1)$~~ $(13+1^2)\%11 = 3$ collision.

$17 \; (13+2^2)\%11 =$

⑦ $36\%11 = 3$ collision

$(36+1)\%11 = 4$ collision

$(36+4)\%11 = 7$ collision

$(36+9)\%11 = \boxed{1}$ ✓

⑧ $24\%11 = 2$ collis.

$(24+1)\%11 = 3$ collision

~~$(24+2^2)\%11 = 7$ collis~~

~~$(24+3^2)\%11 = 5$~~ collision

~~$(24+4)\%11 = 6$~~

~~33 $(24+9)\%11 = 0$ coll~~

~~42 $(24+16)\%11 = 7$ collision.~~

~~49 $(24+25)\%11 = 5$ collision.~~

~~60 $(24+36)\% = 5$ collision.~~

~~73 $(24+49)\%11 =$~~

$(24+4)\%11 = 6$ collision.

$(24+9)\%11 = 0$ collision.

$(24+16)\%11 = 7$ collisi

$(24+25)\%11 = \boxed{5}$ ✓

49

27

5

Hash Table H:

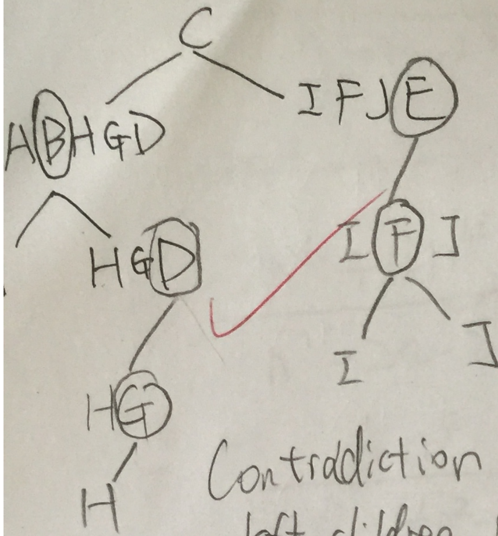
tablesize 11

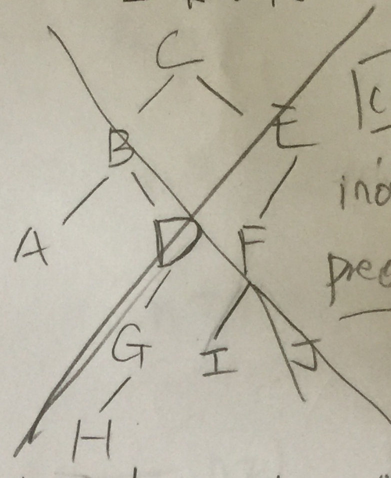
0	22
1	36
2	35
3	25
4	4
5	24
6	13
7	18
8	
9	
10	

6. (10) Construct the unique binary tree corresponding to the given pair of tree traversals if possible. If no such a tree is possible, you must explain/justify your answer clearly for credits.

(a) *Preorder:* C B A D G H E F I J   root L R

Inorder: A B H G D C I F J E   L Root R





inorder : ✓

$\boxed{\text{check}}$: postorder ✗

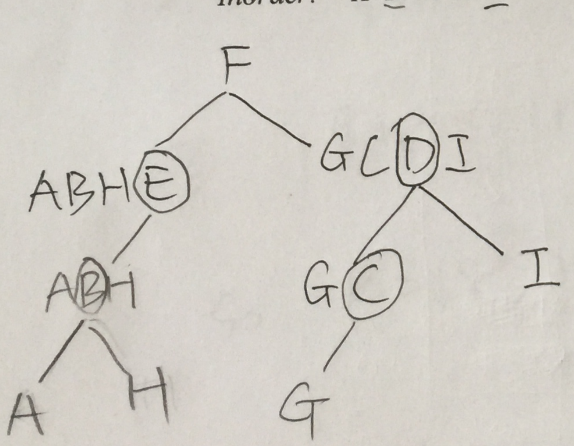
inorder is right, but preorder is wrong!

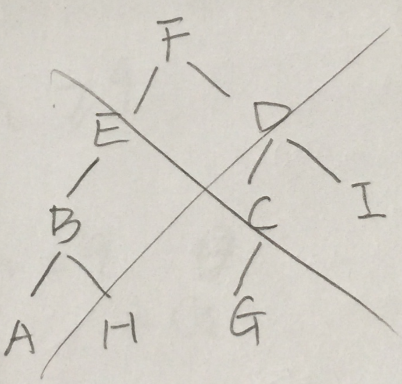
Contradiction in A nodes, when A node to inorder, left children is nothing, but right of A node has children.

$\boxed{\text{No tree!}}$

(b) *Postorder:* H A B E G C I D F

Inorder: A B H E F G C D I





check: inorder ✓

postorder ✗

postorder is wrong

Inorder : In node B, left children is A and right is H, it is a contradiction post of order, H is always in left children in post order

$\boxed{\text{No tree}}$

-7

7. (10) Using the definition of big-O to prove that $\frac{2n^4 - n^3 - 2n^2 + 4}{n^2 - 2n - 27} = O(n^2)$.

$\frac{1}{3}n^2$

$\frac{2n^4 - \cancel{n^3} - \cancel{2n^2} + 4}{n^2 - 2n - 27} \le k n^2$ iff $\exists k > 0, n > 0$ $\forall n_0 > n$

$\cancel{2n^4 - n^3 - 2n^2}$ $n \geq 1$

$\le \frac{2n^4 + 4n^4}{n^2 - 2n - 27}$ $n \geq 1$

$\le \frac{6n^4}{\frac{1}{3}n^2 + (\frac{1}{3}n^2 - 2n) + (\frac{1}{3}n^2 - 27)}$ $n \geq 1$

① $\frac{1}{3}n^2 - 2n \geq 0$

$n(\frac{1}{3}n - 2) \geq 0$

$\frac{1}{3}n - 2 \geq 0$

$n \geq 6$

$\le \frac{6n^4}{\frac{1}{3}n^2}$ $n \geq 9$.

$\frac{1}{3}n^2 - 27 \geq 0$

$n^2 \geq 27 \times 3$

$n^2 \geq 81$

$n \geq 9$

$\le 18n^2$ $n \geq 9$.

So $k = 18$, $n = 9$ to $O(n^2)$.

~~$n \geq 9$~~

You are calculating $\Omega(n^2)$, not $O(n^2)$!

However, your procedures are correct if we assume it is calculating $O(n^2)$.

8. (10) By assuming that all basic operations require the same $\boxed{\text{constant cost } K}$, compute the cost of the resource function $R_w(n)$ in closed-form for the following program segment using the simplified approach as discussed in class. *Remark:* You must first set up a summation equation for $R_w(n)$ and then evaluate the sum(s) clearly for credit.

```
x = 210;
y = 560;
for i = 1 to n*n do
  for j = i to n do
      y = x * y / 660 + 388;
  endfor;
endfor;
```

when $i = n$ first loop, then second loop must be stopped!

~~n keep changing!~~

$$\sum_{i=1}^{n^2} \sum_{j=i}^{n} c$$

$$\Sigma i = \frac{n(n+1)}{2}$$

$$= c \sum_{i=1}^{n^2} \sum_{j=i}^{n} 1$$

$$= c \sum_{i=1}^{n^2} (n-i+1)$$

$$= c \sum_{i=1}^{n^2} [(n+1) - i]$$

$$= c\left((n+1)(n^2) - \frac{n^2(n^2+1)}{2} \right)$$

Check answer!

$$= c \left(n^3 + n^2 - \frac{n^4 + n^2}{\cancel{n^2(n^2+1)}\ 2} \right)$$

= negative number!

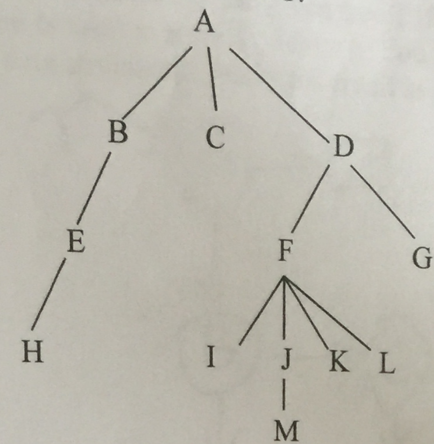
$$Rw(n) = \left(\left(\sum_{i=1}^{n} + \sum_{i=n+1}^{n^2} \right) \sum_{j=i}^{n} \right) K$$

$$= \sum_{i=1}^{n} \sum_{j=i}^{n} K + \sum_{i=n+1}^{n^2} \sum_{j=i}^{n} K$$

$$= K \sum_{i=1}^{n} (n-i+1) + K \sum_{i=n+1}^{n^2} (n-i+1)$$

$$= K\left(n^2 - \frac{n(n+1)}{2} + n\right) + \Theta(n^2) = \Theta n^2$$

9. (12) Consider the following given tree T.

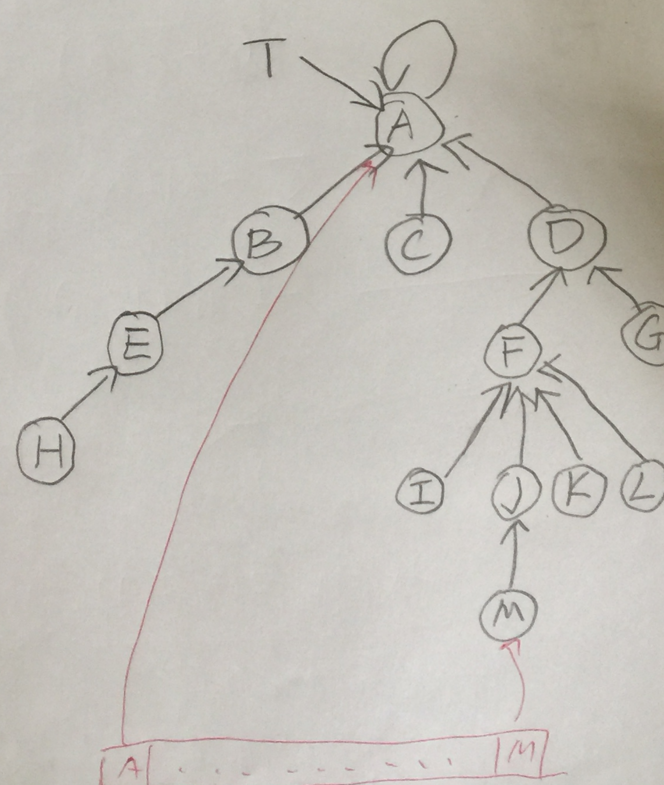


(a) Illustrate the data structure for the given tree T if the parent pointer implementation is used to implement T. *Remark:* You must show all your pointers as well as any auxiliary data structures clearly for credit as discussed in class. You can use alphabetical order to resolve any ambiguity.

+3

→ pointer!

T: start



6

(b) Illustrate the data structure for the given tree T if the leftmost-child list-of-siblings implementation is used to store T. *Remark:* You must show all your pointers as well as any auxiliary data structures clearly for credit as discussed in class.

T: start

NULL:

→ : pointer