<u>**Mid-term Exam Solution**</u>

2016.10.21

1(15 pts) Use the Master Theorem to find the asymptotic solutions for the following recurrences:

(a) $T(n) = 7T(\frac{n}{2}) + n^2$,

(b) $T(n) = T(\frac{n}{2}) + 1$,

(c) $T(n) = 4T(\frac{n}{2}) + n^3$.

Solution:

(a) We have $a = 7$, $b = 2$,

$$f(n)/n^{\log_2 7} = n^2/n^{\log_2 7} = n^{2-\log_2 7} = O(n^{-0.8}),$$

we get Case 1, and thus $T(n) = \Theta(n^{\log_2 7})$.

(b) We have $a = 1$, $b = 2$,

$$f(n) = \Theta(1) = \Theta(n^{\log_2 1}) = \Theta(1),$$

we get Case 2, and thus $T(n) = \Theta(\log_2 n)$.

(c) We have $a = 4$, $b = 2$,

$$f(n)/n^{\log_2 4} = n^3/n^2 = n,$$

we get Case 3, and thus $T(n) = \Theta(n^3)$. Note that the regularity condition is satisfied as $4(n/2)^3 = n^3/2 \le cn^3$ for $c = 1/2$.

2(10 pts) Heap and hashing are two well-known data structures for maintaining a dynamic set of elements. Describe an application scenario where you should use heap instead of hashing, and another scenario where you should use hashing instead of heap.

sol) If the elements obey an ordering relationship, and we want to find min or max, we should use heap. If we need to search for any element, not just min or max, we should use hash.

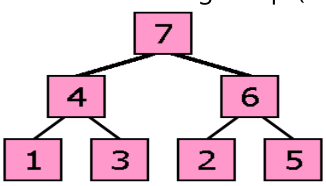
3(15 pts) Suppose we start with an empty Maxheap, and insert each of the following items, in order: 1 2 3 4 5 6 7 8

(a) Draw the resulting Heap (as a tree).

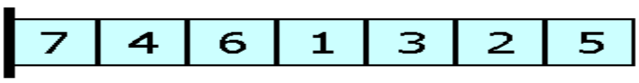
(b) Draw the array representation of the above Heap.

(c) Now, we perform one Heap delete operation on the Heap from the previous parts. Draw the resulting Heap (as a tree)

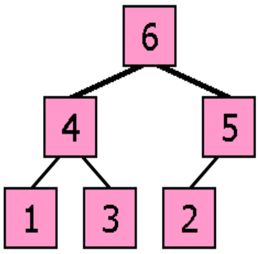
a)



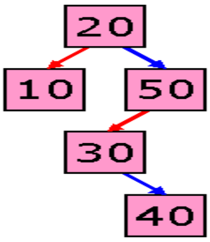
b)



c)



4(20 pts) A Binary Search Tree is shown below.



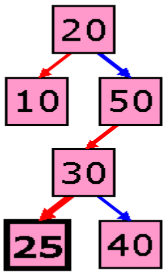
Perform each of the following operations and show the resulting tree. <u>For each part, start with the original tree above, not the tree resulting from earlier parts.</u>

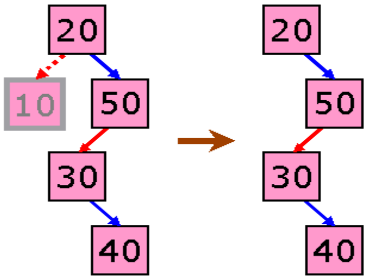
(a) insert 25.

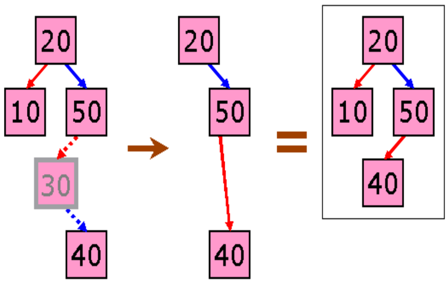
(b) delete 10.

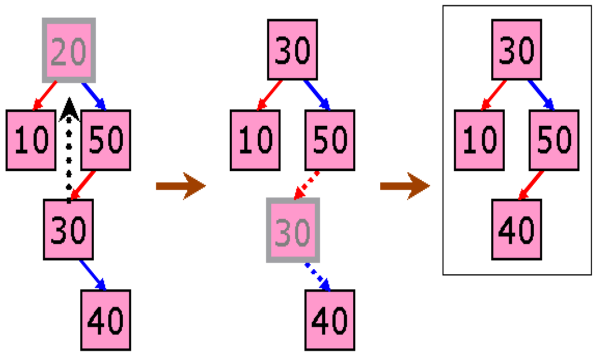
(c) delete 30.

(d) delete 20









5(20 pts) Given an initially empty hash table with 13(indexes 0~12) and hash function H (x) = x % 13, insert these values in the order given: 5, 17, 4, 22, 31, 43, 44.

(a) separate chaining

(b) linear probing

(c) quadratic probing

(d) Double hashing with H’ (x) = 7 – (x % 7)

a. Separate chaining

0	
1	
2	
3	
4	**17,4,43**
5	**5,31,44**
6	
7	
8	
9	**22**
10	
11	
12	

b. Linear probing

0	
1	
2	
3	
4	**17**
5	**5**
6	**4**
7	**31**
8	**43**
9	**22**
10	**44**
11	
12	

c. Quadratic probing

0	**43**
1	**44**
2	
3	
4	**17**
5	**5**
6	**31**
7	
8	**4**
9	**22**
10	
11	
12	

d. Double hashing with $H'(x) = 7 - (x \% 7)$

0	**31**
1	
2	**44**
3	
4	**17**
5	**5**
6	
7	**4**
8	
9	**22**
10	**43**
11	
12	

6(20 pts)

(a) Quick sort algorithm consists of three fundamental part.

Write Quick sort algorithm in pseudo-code

sol) refer to Lecture Notes

(b) What is the worst case of quick sort algorithm in big Oh notation?

Explain what case the worst case is.

sol) $O(n^2)$

Keys are sorted in ascending or descending order.

(c) What is the disadvantage of Merge sort?

Sol) need extra storage (O(n)) since not in-place