CPE206 Algorithms

Midterm Exam, Duration: 70+5 mins.

Submit your answers in a pdf file (other formats are not allowed)

Do not send e-mail.

Q1.

(a) [10P] What is the last content of the array A = [-13, 10, -11, 6, -7, 4], if the following function **noname** is called with the input parameters (A, 6). Explain what the algorithm does generally. Find its running time.

```
noname (A,n) // A[1…n]
    i = 1
    j = n
    while i<=j
        if A[i]<0
            i = i+1
        else
            swap (A[i], A[j])
            j = j-1
```

n

$T(n) = \Theta(n)$

1	2	3	4	5	6
-13	10	-11	6	-7	4
	i				J
-13	4	-11	6	-7	10
	i			J	
-13	-7	-11	6	4	10
			J	i	

(b) [10P] Convert **noname** to a recursive function.

```
noname_rec(A, p, q)
    if p ≤ q
        if A[p] < 0
            noname_rec(A, p+1, q)
        else
            swap(A[p], A[q])
            noname_rec(A, p, q-1)
```

Q2 [20P]: Suppose that the following recurrences belong to three algorithms for solving the same problem. Solve the recurrences by the Master method. Order them from the best algorithm to the worst.

i. $T(n) = 9T(n/3) + \Theta(n)$
ii. $T(n) = 8T(n/2) + \Theta(n^4) \rightarrow f$
iii. $T(n) = 2T(n/4) + \Theta(\sqrt{n})$

i) Compare n with $n^{\log_3 9} = n^2$

$n = \Theta(n^{2-\varepsilon}) \qquad \varepsilon = 1$

Case 1: $T(n) = \boxed{\Theta(n^2)}$

ii) Compare f n^4 with $n^{\log_b a} = n^{\log_2 8} = n^3$

$n^4 = \Omega(n^{3+\varepsilon}) \qquad \varepsilon = 1$

Reg. Cond. $af(n/b) \le c \cdot f(n)$

$8 \cdot f\left(\frac{n}{2}\right) \le c \cdot f(n)$

$8 \cdot \left(\frac{n}{2}\right)^4 \le c \cdot n^4$

$8 \cdot \frac{\cancel{n^4}}{16} \le c\cancel{n^4}$

$\frac{1}{2} \le c \qquad$ Let $c = \frac{1}{2} < 1$

Case 3: $T(n) = \boxed{\Theta(n^4)}$

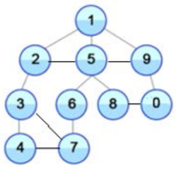
iii) Compare $\sqrt{n}$ with $n^{\log_4 2} = n^{1/2}$

$\sqrt{n} = \Theta(\sqrt{n})$

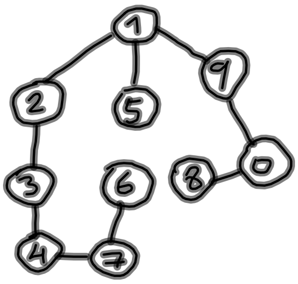
Case 2: $T(n) = \boxed{\Theta(\sqrt{n}\lg n)}$

$\sqrt{n}\lg n < n^2 < n^4$

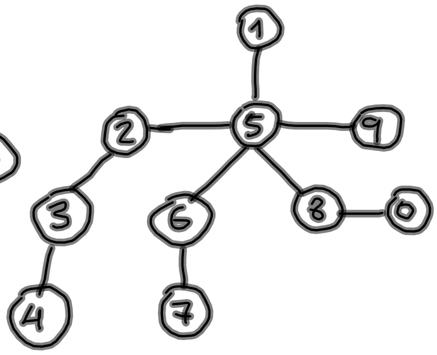
Q3 [20P]: Traverse the following graph, starting from a vertex with the last digit of **your student number**, by Depth First Search and Breadth First Search algorithms. Make sure that when there are multiple nodes to be considered, the smallest one will be selected. Draw, separately, the result of each algorithm.



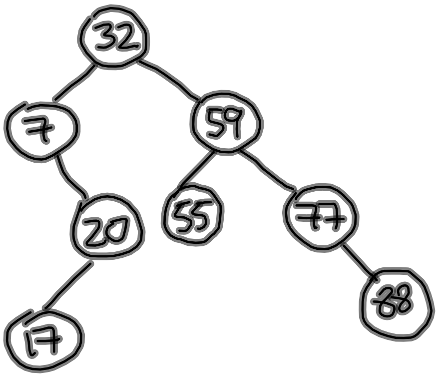
DFS



BFS



Q4 [10P]: For a given set {32,59,7,77,20,17,55,88}, build a binary search tree containing these numbers (insert the numbers one by one into an empty BST) and then perform the **preorder** tree walk. Draw only one tree. Do not draw a new tree at each insertion.



root left right

32, 7, 20, 17, 59, 55, 77, 88

Q5 [10P]: Given an array A[1...n] and a method **doit(A, i, j)** which reverses the order of elements in A between positions i and j (both inclusive). What happens to A after the following there calls for **1 < k <= n** ?

doit (A, 1, k);
doit (A, k + 1, n);
doit (A, 1, n);

i j

A= [6,1,7,2,5,3,8,10,11,4]

10,8,3,5,2

A: 1 2 ... k-1 k k+1 . . n

k, k-1, ... 2 1 k+1 . . . . n

k k-1 ... 2 1 n n-1 ... k+1

k+1 k+2 ... n-1 n 1 2 ... k-1 k

Q6 [12P]: Fill the blanks in the following statements

a. The worst case of the Quick sort occurs when …sorted or reverse sorted…

b. …Quick… and …merge… sorting algorithms are designed by divide and conquer paradigm.

c. The worst case running time of the …quick insertion… sort is $O(n^2)$

d. Quick sort and insertion sort algorithms sort the numbers …in place…

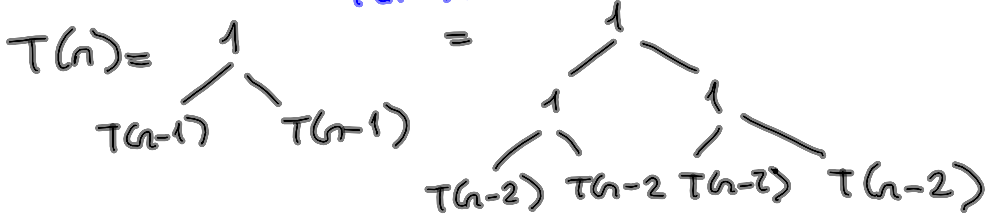
e. …DP… solves every subproblem just once and stores the answer in a …table…

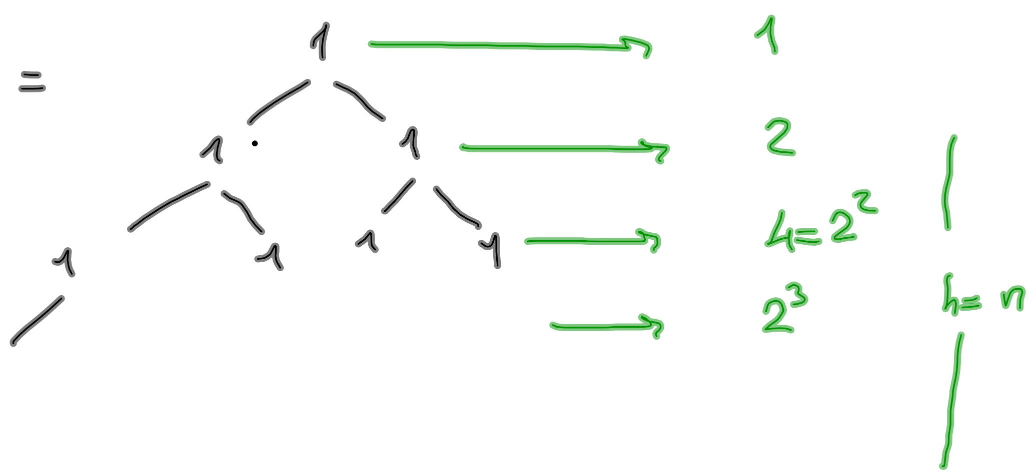
f. The subproblems must be dependent to use …DP…, however, the subproblems may not be dependent to use …D&C…

Q7 [10P] Solve the following recurrence using the recursion tree method.

T(n) = 2T(n-1)+ 1

$T(n-1) = 2T(n-2)+1$





$\Theta(1)$ . . . . . . $\Theta(1)$ . . . 2^n

+

$$\underbrace{1+2+2^2+\ldots+2^n}_{\text{Geometric series}} = \frac{2^{n+1}-1}{2-1}$$

$$\boxed{T(n) = \Theta(2^n)}$$