DAA - Unit Test-01

(18CS42)

Shahul Hameed. S
1KN18CS097
CSE 'A' Sec 4th Sem

(1) What is an algorithm? Explain the properties of an Algorithm? Explain the notion of an algorithm with an Eg.

> **Algorithm:** An algorithm is a finite sequence of unambiguous instructions to solve a particular problem.

Properties of an algorithm:

* **Input:** Zero / more quantities are externally supplied.
* **Output:** At least one quantity is produced.
* **Definiteness:** Each instruction is clear and unambiguous. It must be perfectly clear what should be done.
* **Finiteness:** If we trace out the instruction an algorithm, then for all cases, the algorithm terminates after a finite number of steps.
* **Effectiveness:** Easy instruction must be very basic so that it can be carried out in principle. It is not enough that each operation be definite as in criterion c, it must ~~be~~ also be feasible.

Notion of an algorithm: The non-ambiguity requirement for each step of an algorithm can't be compromised.

problem
↓
algorithm
↓
I/P → [computer] → O/P

The range of I/P for which an algorithm works has to be signified carefully. The same algorithm can be represented in several different ways. Several algorithms for solving the same problem may exist.

Eg 1: Euclid's algorithm gcd(m, n)

Step 1: If n=0, return the value of m as the answer and stop. otherwise proceed to step 2.

Step 2: Divide m by n and assign the value of the remainder to r.

Step 3: Assign the value of n to m and the value of r to n. Go to step 1.

(2) P.T (i) $\frac{1}{2} n(n-1) \in \Theta(n^2)$

(ii) $n! \in \Omega(2^n)$

(i) if $n=8$

$\frac{1}{2} n(n-1) = \frac{1}{2} * 8(8-1) = 28$

$n^2 = 8^2 = 64$

If $n=4$

$\frac{1}{2} n(n-1) = \frac{1}{2} * 4(4-1)$

$n^2 = 4^2 = 16$

All such comparisons indicate the that $\frac{1}{2} n(n-1) < n^2$

$\therefore \frac{1}{2} n(n-1) \in \Theta(n^2)$ $\therefore c_2 g(n) \leq f(n) \leq c_1 g(n)$

(ii) $n! \in \Omega(2^n)$

if $n=2$

$f(n) = 2! = 2$

$g(n) = 2^n = 2^2 = 4$

If $n=5$

$f(n) = 5! = 120$ and

$g(n) = 2^n = 2^5 = 32$

$f(n) \geq c * g(n)$ //.

DAA - unit Test - 02

(18CS42)

Shahul Hameed . S
1KN18CS097
CSE 'A' sec
4th sem

① with Suitable Example Explain the Significance of order of growth in analysing Algorithms.

→ Measuring the performance of an algorithm in relation with the input size is called order of growth.

Significance:- All the exponential belong to the same order of growth regardless of the base of the exponent. Exponential functions grow very quickly, so Exponential algorithms are only useful for small problems.

Similarly for the log terms, the base of log doesn't matter, changing bases ~~is~~ is the equivalent of multiplying by a constant, which doesn't change the order of growth.

Eg:-

n	$\log n$	$n \log n$	n^2	2^n
1	0	0	1	2
2	0	2	4	4
4	2	8	16	16
8	3	24	64	256
16	4	64	216	65,536.

from the above table log fn is the slowest growing function and the exponential function 2^n is fastest and grows rapidly with varying i/p size n.

(Q) Compare order of growth of

(i) n^3 and 2^n

(ii) $\frac{1}{2} n (n-1)$ and n^2

> (i) $f(n) = 2$ $\qquad$ $g(n) = 2^n$

$n^3 = 2^3 = 8$ $\qquad$ $n = 2 \Rightarrow g(2) = 2^2 = 4$

$f(n) \geq c * g(n)$

$\therefore n^3 \in \Omega(2^n)$

(ii) $f(n)$ and $g(n)$, $n = 2$

$n = 2$, $n^2 = 2^2 = 4$

$\frac{1}{2} \cdot 2 (2-1)$

$f(n) = 1$

if $n = 3$

$n = 3$ $\qquad$ $n^2 = 3^2 = 9$

$\frac{1}{2} \, 3(3-1)$

$\frac{1}{2} \, 3(2) = 6/2 = 3$

$f(n) \leq c * g(n)$

$\therefore \frac{1}{2} n(n-1) \in \Theta(n^2)$

DAA - unit Test =03

(18CS42)

Shahul Hameed . S
1KN18CS097
CSE 'A' sec
4th sem

(1) write an algorithm for sorting the numbers using Quick Sort Derive best case, worst case and average case efficiency of an algorithm

> Quick sort (arr[], low, high)

```
{
  if (low < high)
    pi = partition (arr, low, high);
    quicksort (arr, low, pi-1);
    quick sort (arr, pi+1, high);
}
```

<u>worst case</u>:- $T(n) = T(0) + T(n-1) + \theta(n)$

$T(n) = T(n-1) + \theta(n)$

$\therefore \theta(n^2)$ is worst case efficiancy

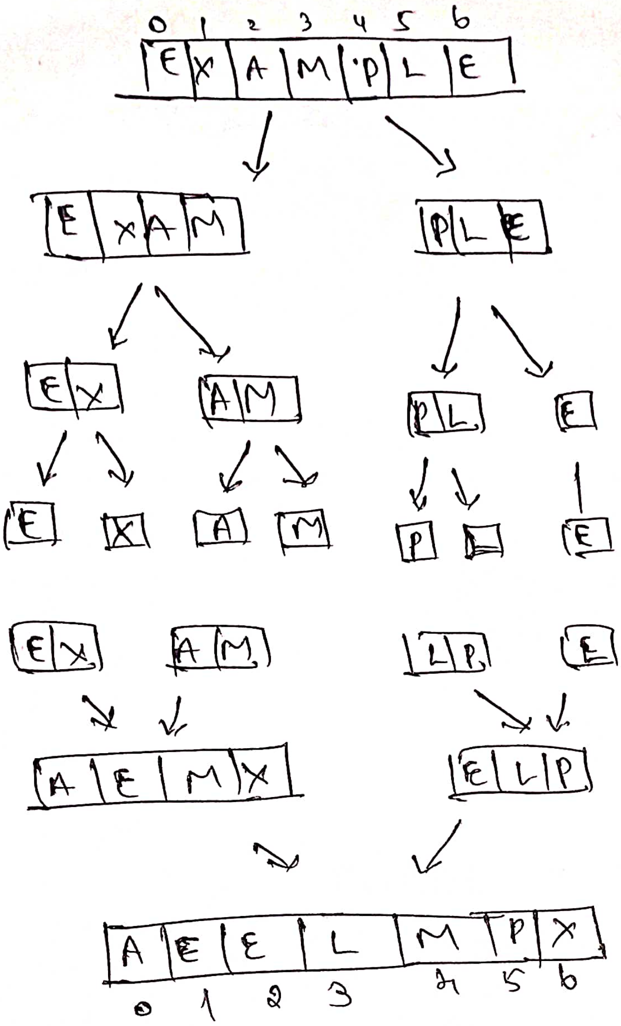
Best Case :- $T(n) = 2T(n/2) + \theta(n)$

$\therefore \theta\, n \log n$ - best case efficiency.

Average <u>case efficiency</u> :- $T(n) = T(n/9) + T(9n/10) + \theta(n)$

$\therefore \theta(n \log n)$ is average case efficiency.

Sort the list E, X, A, M, P, L, E in alphabetical order using the Quick sort.



DAA - unit Test - 04

(18CS42)

Shahul Hameed. S
1KN18CS097
CSE 'A' sec
4th sem

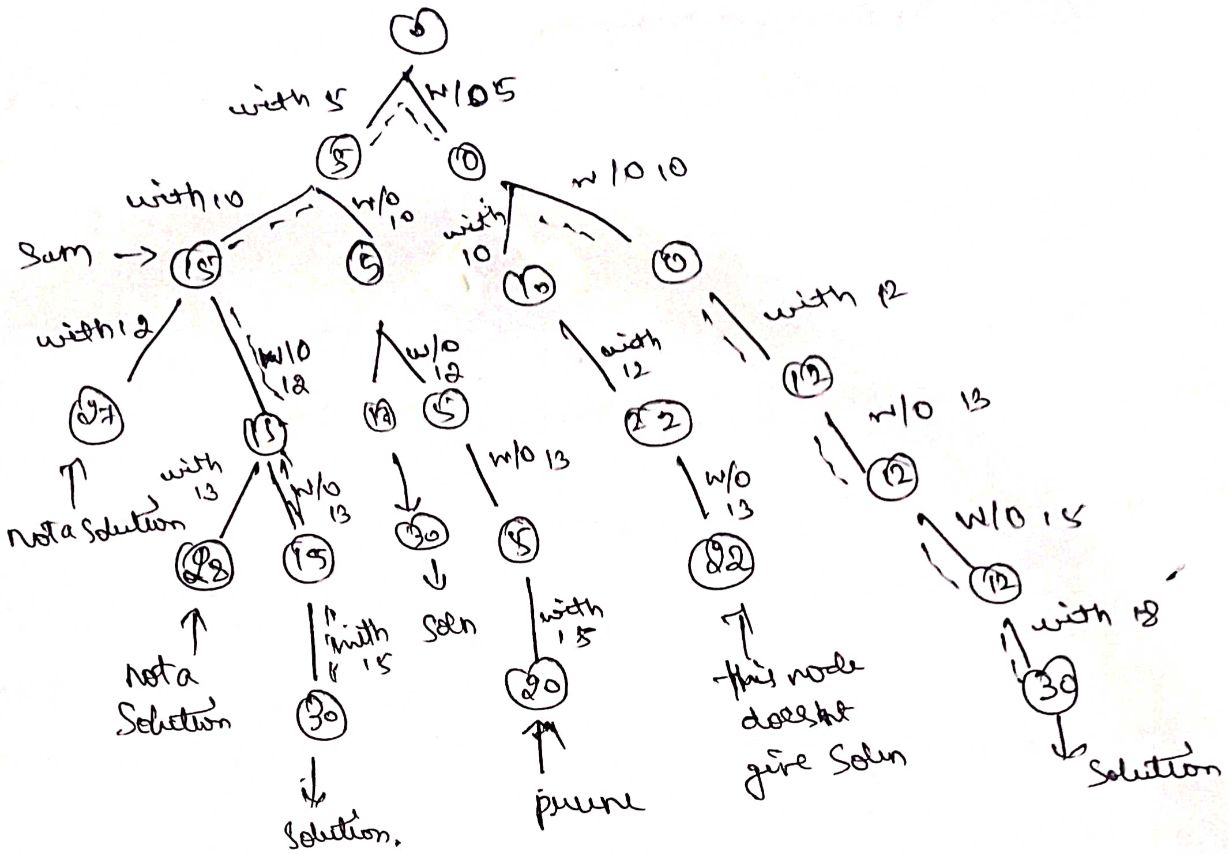
(1) Apply backtracking to solve Subset Sum problem for the instance $n = 6$, $d = 30$, $S = \{5, 10, 12, 13, 15, 18\}$

> Initially Subset $= \{\}$ Sum $= 0$

5	5	Then add next element
5, 10	15 ∵ 15 < 30	Add next element
5, 10, 12	27 ∵ 27 < 30	Add next element
5, 10, 12, 13	40	Sum exceeds $d = 30$
5, 10, 12, 15	42	Sum exceeds $d = 30$
5, 10, 12, 18	45	∴ Back track
5, 10		
5, 10, 13	28	not feasible ~~backtrack~~
5, 10, 13, 15	33	
5, 10		
5, 10, 15	30	Solution obtained as Sum $= 30 = d$

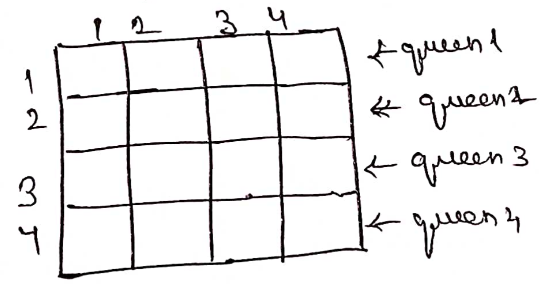
The state space tree can be drawn as follows,

$\{5, 10, 12, 13, 15, 18\}$

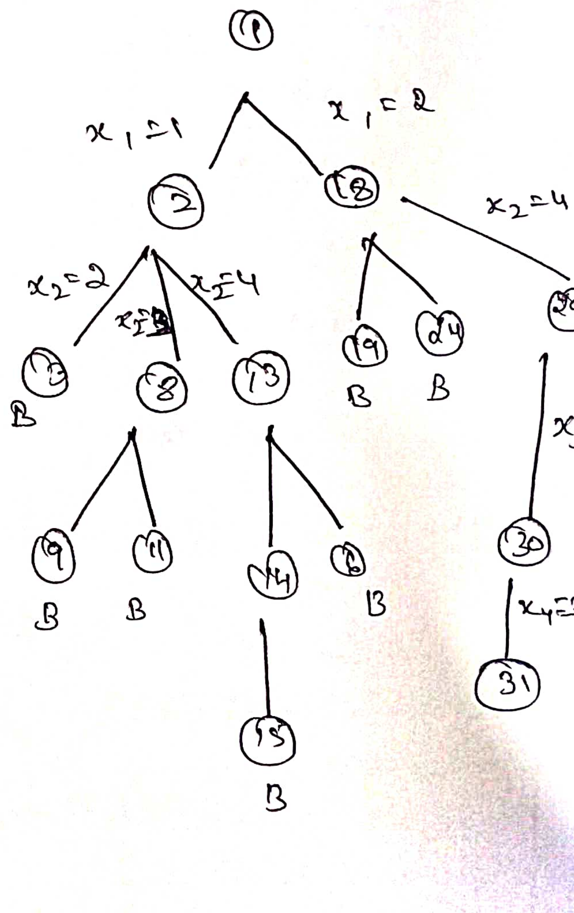


Q2 Draw a state space tree to generate solution to 4-Queen problem.

> Board for the 4-queen problem.



In the following state space tree x denotes an unsuccessful attempts to place a queen in the indicated column. The numbers above the nodes indicate the order in which the nodes are generated.



DAA - unit Test - 05

(18CS42)

Shahul Hameed. S
1KN18CS097
CSE 'A' sec
4th Sem

(1) what is Brand Bound algorithm? How it is different from back tracking.

7 Brand and bound is a algorithm design paradigm. which is generally used for solving Combinational optimization in terms of time complexity and may require exploiting all possible permutations in worse case. The Branch and Bound algorithm technique solves, these problems relatively quickly.

Back Tracking	Branch and bound algorithm
* Backtracking traverses the state space tree by DFS manner.	* Brand and bound traverse the tree in any manner BFS/DFS.
* Backtracking involves feasibility function.	* Branch and bound involves a bounding function.
* Backtracking is used for solving decision problem.	* Branch and bound is used for solving optimization problem.
* Backtracking is more efficient	* Branch and bound are less efficient.

(Q2) what is Hamiltonian cycle? Give the backtracking based algorithm to find the hamiltonian cycle in graph. Write the functions used to generating next vertex and for finding hamiltonian cycle.

> hamiltonian cycles:- A path through a graph that starts and ends at the same vertex and includes every exactly once. Also known as tour.

Algorithm {

```
repeat
{
 next vertex (k);
  if [x[k] = 0] then
  return ;
 if (k=n) then
  write (x[1:n])
 Else
   H cycle (k+1);
  } until (false);
 }
```

Algorithm: next value (k)

```
{
 repeat
  {
   x[k] := (x[k]+1) mod (m+1);
    if (x[k]=0) then return;
    for j = 1 to n do
    {
     if ((G[k,j] ≠ 0) and (x[k] = x[j]))
     then break;
    }
    if (j = n+1) then return;
   } until (false);
}
```