CSE - 207 (2011-12)

Section B

10. (a)

Greedy Problems:- ~~Greedy probs~~ In case of this method, only the option that currently seems to be the best is chosen.

In other words, if a problem can be divided into subtasks, then only the ~~subs~~ subtask that currently seems to be the best option is chosen, without taking into consideration the next subtasks. After an option is chosen, no backtrack is possible.

Divide & Conquer Problems:- Such problems can be broken down into smaller subproblems of the same type. Since the subproblems are of the same type, the recursive function can be called to solve these problems.

In such cases, backtracking is done quiet frequently when recursive calls return values.

Dynamic Programming:- Such problems can be broken down into overlapping subproblems and they have optimal substructure. The solutions to the subproblems are stored (typically in a table, no storing is done in divide and conquer) for later use.

This method greatly reduces time complexity for problems that would take a long time to solve using divide and conquer.

10 (b)

Out of syllabus

10(c).

Let's assume Z contains x_m

Observe that, the letter Z_K is the last common letter of X and Y. Since, we have assumed that Z contains x_m, then the last common letter of X and Y(Z_K) has to be x_m because there's no letter in X after x_m.

$\therefore \quad x_m = Z_K$

but from the question $x_m \neq Z_K$, this is a contradiction.

$\therefore$ Z cannot contain x_m, So, Z must be the LCS of X_{m-1} and Y

10(d)

	empty	T	G	C	A	T	A
empty	0	0	0	0	0	0	0
A	0	0	0	0	1	1	1
T	0	(1)	1	1	1	2	2
C	0	1	1	(2)	2	2	2
T	0	1	1	2	2	(3)	3
G	0	1	2	2	2	3	3
A	0	1	2	2	3	3	(4)
T	0	1	2	2	3	4	4

From BackTracing:

LCS ⇒ TCTA

Length = 4

11. (a) - out of syllabus.

11. (b) ~~In case of Greedy algorithm~~

For the Fractional Knapsack problem, you can choose a fraction of an item, so that the total weight doesn't exceed the weight limit. So you can ~~start~~ Keep by choosing the items with the largest (value/weight) ratio, since the option to choose fractions ensures that weight limit won't be exceeded. So. In other words, you can ~~keep choosing~~ choose the greedy option at every step,

But for the 0/1 problem, the ~~weigh~~ greedy option can't always be chosen, since there's a chance that the weight limit will be exceeded. So, greedy method won't work here.

11 (c)

	0	1	2	3	4	5	6
0	0	0	0	0	0	0	0
1	0	0	50	50	50	50	50
2	0	0	50	50	50	80	80
3	0	12	50	62	62	80	92
4	0	12	50	62	62	95	107

items:

	W	V
1	2	50
2	3	30
3	1	12
4	3	45

From Backtrace:- items selected 1, 3 & 4

maximum value = 107

12, 13 are out of syllabus