Homework 3　林業誠　E24105038　114資訊甲

1)

		0	1	0	1	1	0	1	1	0
	0	0	0	0	0	0	0	0	0	0
1	0	0↑	1↖	1←	1↖	1↖	1←	1↖	1↖	1←
0	0	1↖	1↑	2↖	2←	2←	2↖	2←	2←	2↖
0	0	1↖	1↑	2↖	2↑	2↑	3↖	3←	3←	3↖
1	0	1↑	2↖	2↑	3↖	3↖	3↑	4↖	4↖	4←
0	0	1↖	2↑	3↖	3↑	3↑	4↖	4↑	4↑	5↖
1	0	1↑	2↖	3↑	4↖	4↖	4↑	5↖	5↖	5↑
0	0	1↖	2↑	3↖	4↑	4↑	5↖	5↑	5↑	6↖
1	0	1↑	2↖	3↑	4↖	5↖	5↑	6↖	6↖	6↑

$\therefore$ $\boxed{LCS = \langle 1,0,0,1,1,0 \rangle}$ #

---

2) (a) Stable sorting algorithm is that sorts the equivalent elements in the same order in output as in input and stable means that the equivalent elements will be in the same order in output as in input.

(b) It is because counting sort iterates through the input array in reverse order in the last loop when placing elements in output array, it ensures that the elements with equal keys are placed in output array in the same relative order as in the input array. This makes counting sort a stable sorting algorithm.

---

3) ↳ array with length n

set max. and min. on the first element

⇒ if n is odd, # of comparison $= 3\left(\frac{n-1}{2}\right)$ $\boxed{= \frac{3}{2}n - \frac{3}{2}}$

⇒ if n is even, # of comparison $= 3\left(\frac{n-2}{2}\right) + 1$ $\boxed{= \frac{3}{2}n - 2}$

set min. to the smaller element and set max. to the larger element (compare 1 time)

$\therefore$ The time complexity is at most $\boxed{\Theta(n)}$ #

---

4) # of comparisons $\boxed{\leq 3\left\lfloor \frac{n}{2} \right\rfloor}$

$\therefore$ Compare elements at most $\boxed{3\left\lfloor \frac{n}{2} \right\rfloor}$ times #

7)

① If the number of possible lengths for the rod pieces is very small (eg 2, 3), just directly solve the problem by enumerating all possible combinations of length and select the one with maximum value.

② If the length of the rod is very small (eg 1, 2), solve the problem by directly computing the value of the rod pieces that can be obtained.

③ If the problem has some additional constraints or assumptions (eg. all rod pieces must have equal lengths, the total length of the rod must be a multiple of a certain value), it may be possible to exploit these constraints to simplify the problem and obtain a direct solution.

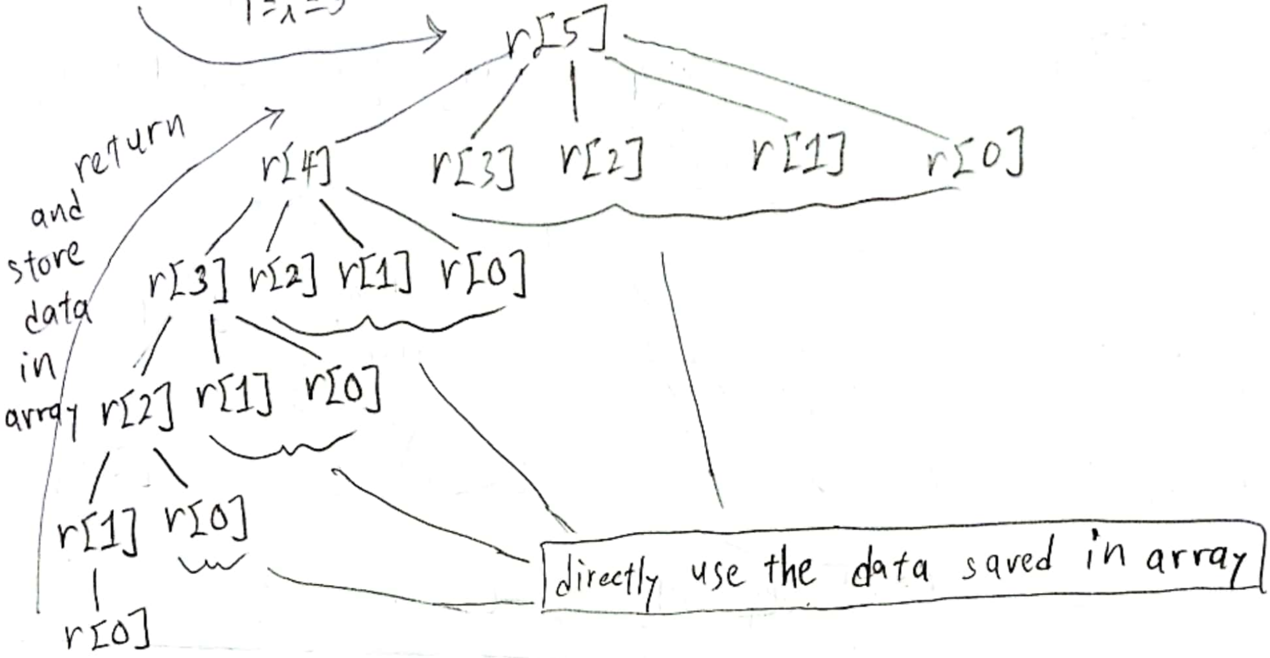
8)

⇒ Memorized-cut-rod (p, 5)

↳ initialize r array as $-\infty$

↳ return Memorized-cut-rod-aux (p, 5, r)

$$\Rightarrow r[5] = \max_{1 \le i \le 5} \left[p[i] + \text{Memorized-cut-rod-aux}(p, 5-i, r) \right]$$



$\Rightarrow r[0] = 0$

$\Rightarrow r[1] = p[1] + r[0] = 1$

$$\Rightarrow r[2] = \max_{1 \le i \le 2} (p[i] + r[2-i]) = \max(2, 5) = 5$$

$$\Rightarrow r[3] = \max_{1 \le i \le 3} (p[i] + r[3-i]) = \max(6, 6, 8) = 8$$

$$\Rightarrow r[4] = \max_{1 \le i \le 4} (p[i] + r[4-i]) = \max(9, 10, 9, 9) = 10$$

$$\Rightarrow r[5] = \max_{1 \le i \le 5} (p[i] + r[5-i]) = \max(11, 13, 13, 11, 10) \boxed{= 13} \#$$

∴ The maximal revenue r_5 is $\boxed{13}$ #

9) a) F, dynamic programming is a technique

b) F, ⇒ divide n elements into groups of 5,

↳ $T(n) \le T(\lceil \frac{n}{5} \rceil) + T(\frac{7}{10}n) + O(n)$

$\Rightarrow T(n) \le c\lceil \frac{n}{5} \rceil + c(\frac{7}{10}n) + an$

$\le \frac{cn}{5} + c + \frac{7}{10}cn + an = \frac{9}{10}cn + c + an = cn + (-\frac{cn}{10} + c + an)$

$\le cn$, if $c \ge \frac{10an}{n-10}$

$\boxed{\therefore T(n) = O(n)}$

⇒ divide n elements into groups of 3,

↳ $T(n) \le T(\lceil \frac{n}{3} \rceil) + T(\frac{2}{3}n) + O(n)$

$\Rightarrow T(n) \le c\lceil \frac{n}{3} \rceil + c(\frac{2}{3}n) + an$

$\le \frac{cn}{3} + c + \frac{2}{3}cn + an = cn + c + an \not\le cn$ $\boxed{\therefore T(n) \ne O(n)}$

$\boxed{\therefore \text{They do not give the same linear time complexity}}$ #

c) F, the optimal substructure property means that an optimal solution to a problem can be found by combining optimal solutions to its subproblems. However, just because a problem satisfies the optimal substructure property does not guarantee that every locally optimal solution is also globally optimal solution.

10)

		D	F	A	D	B	D	C	E	B	E	A	G
	0	0	0	0	0	0	0	0	0	0	0	0	0
E	0	0↑	0↑	0↑	0↑	0↑	0↑	0↑	1↖	1←	1↖	1←	1←
F	0	0↑	1↖	1←	1←	1←	1←	1←	1↑	1↑	1↑	1↑	1↑
B	0	0↑	1↑	1↑	1↑	2↖	2←	2←	2←	2↖	2←	2←	2←
E	0	0↑	1↑	1↑	1↑	2↑	2↑	2↑	3↖	3←	3↖	3←	3←
G	0	0↑	1↑	1↑	1↑	2↑	2↑	2↑	3↑	3↑	3↑	3↑	4↖
E	0	0↑	1↑	1↑	1↑	2↑	2↑	2↑	3↖	3↑	4↖	4←	4↑
C	0	0↑	1↑	1↑	1↑	2↑	2↑	3↖	3↑	3↑	4↑	4↑	4↑
A	0	0↑	1↑	2↖	2←	2↑	2↑	3↑	3↑	3↑	4↑	5↖	5←
E	0	0↑	1↑	2↑	2↑	2↑	2↑	3↑	4↖	4←	4↖	5↑	5↑
B	0	0↑	1↑	2↑	2↑	3↖	3←	3↑	4↑	5↖	5←	5↑	5↑
D	0	1↖	1↑	2↑	3↖	3↑	4↖	4←	4↑	5↑	5↑	5↑	5↑
A	0	1↑	1↑	2↖	3↑	3↑	4↑	4↑	4↑	5↑	5↑	6↖	6←

a) ∴ The maximum number of ships which can leave at 8am is $\boxed{6}$ #

b) $\boxed{\langle F, B, C, E, B, A \rangle \ \#}$ = LCS