50.004 – Homework 11
Ashlyn Goh 1002840 Cl02 Nov 2018

Q1.
True.
When dynamic programming is applied, each subproblem is only solved once. However, other methods that do not take advantage of overlapping subproblems may re-compute the value of the same sub-problem many times. Hence, dynamic programming takes far less time in finding a solution since it avoids the re-computation of the answer of subproblems.

Q2
(D) Time complexity decreases and space complexity increases
When we use the top-down approach of dynamic programming, we would use memoisation to store values of sub-problems in a table, avoiding the need to compute them twice. Since we do not have to do as many computations, the time complexity decreases. However, we need to store the values of the sub-problems so the space complexity increases.

Q3
Modified code (line 4 to 6)

```
4|    q = p[j]
5|    for i=1 to j-1
6|          q = max(q, p[i] + r[j-i] - c)
```

- Modification to line 4 (q = p[j]) accounts for the case where there are no cuts. When i=j, we should make no cuts so the total revenue from this case is just p[j].
- Modification to line 5 (for loop to j-1 instead of j previously) is because we have already taken into account the case when i=j.
- Modification to line 6 is merely just reflecting the cost of making a cut in the total revenue calculated.

Q4
Recursion on its own (without memoisation) often involve doing more work than necessary where the algorithm would repeatedly solve the common sub-sub problems.

With dynamic programming, recursion is done with memoisation (top-down) or via bottom-up approach. It avoids the work of re-computing the answer every time it solves each sub-sub problem.

Hence, dynamic programming does not implement the recursion equations directly because doing so would take more computation time.

50.004 – Homework 11
Ashlyn Goh 1002840 Cl02 Nov 2018

Q5 (i)
Recursive formula:
$for\ n > 3, P(n) = P(n-1) + P(n-2) + P(n-3)$
else P(1) = 2, P(2) = 4, P(3) = 7

From the hint given, each binary string with n>3 has either one of the 3 properties given.
For property (a), the number of binary strings satisfying the property is P(n-1).
For property (b), the number of binary strings satisfying the property is P(n-2).
For property (c), the number of binary strings satisfying the property is P(n-3).
Hence, we have P(n) = P(n-1) + P(n-2) + P(n-3) when n>3.

Q5 (ii)
The running time of the dynamic programming algorithm is O(n).