1) Write an algorithm that given an unsorted array returns whether an increasing subsequence of length 3 exists or not in the array. Your algorithm should run in O(n) time complexity and O(1) space complexity.

15 points

2) Consider the following multithreaded algorithm for performing pairwise multiplication on *n*-element arrays A[1..n] and B[1..n], storing the multiplications in C[1..n]:

```
MUL-ARRAYS(A, B, C)
1   parallel for i = 1 to A.length
2         C[i] = A[i] * B[i]
```

Analyze the work, span and parallelism of this algorithm.

15 points

3) Design a dynamic programming algorithm for the version of the knapsack problem in which there are unlimited quantities of copies for each of the n item kinds given. Indicate the time efficiency of the algorithm.

20 points