**Q1**: Consider the following knapsack problem:

|  |  |  |
| --- | --- | --- |
| Item | Value | Weight |
| 1 | 1 | 1 |
| 2 | 3 | 2 |
| 3 | 4 | 1 |
| 4 | 2 | 2 |
| 5 | 3 | 3 |
| 6 | 6 | 2 |

The capacity of the knapsack is 9. Use the DP algorithm to solve it.

**Q2:** Given 8 jobs with the following (v, s, f)-values (v=value, s= start time, and f= finish times): a=(4,0,5), b=(5,6,9), c=(6,5,8), d=(5,3,6), e=(4,5,7), f=(12,8,11), g=(2,7,10), h=(7,9,13).

Use a DP algorithm to find a set of mutually compatible jobs with the maximal total value.

**Q3** Lisa will graduate next year, and she wants to find a good job, and build a career path. An ideal career path to her is that the salaries are never decreasing, and ideally, are always multiplying. One day, Lisa met a fortune teller, and was told a sequence of jobs to choose. Lisa has not taken CS4335, and she asked your help to design a method to choose the jobs. Again, we have formulated the problem formally.

A is a sequence of positive integers (represents the salaries): A = (a1, a2, ... an). A **multiplication subsequence of** A is a subsequence S=() satisfies that (1) S is obtained by remove some entries of **A** sequentially; that is, i1<i2<i3<...<ik, is in **A**; and (2) is a multiple of , is a multiple of , …; that is, if we let divide by , ,the remainder is zero.

For example, if A =(1, 2, 3, 3, 4, 5, 6, 7, 8, 15), then (1, 2, 4, 8), (1, 3, 3, 6), and (1, 3, 15) are multiplication subsequences of A.

1. Given A as a sequence of positive integers, design an algorithm to identify a longest multiplication subsequence.
2. Define the weight of a sequence as the sum of the elements in the sequence. Design an algorithm to identify a maximum weighted multiplication subsequence.