```
CS 519-005, Algorithms (MS/MEng-level), Winter 2018
HW5 - DP (part 1: simple)

HWs 5-7 are all on DPs.

Due Monday Feb 12, 11:59pm.
No late submission will be accepted.

Need to submit report.txt, mis.py, bsts.py, bitstrings.py.
mis.py will be graded for correctness (1%).

Textbooks for References:
[1] CLRS Ch. 15
[2] KT Ch. 6, freely available online (strongly recommended!):
    http://www.aw-bc.com/info/kleinberg/assets/downloads/ch6.pdf

[UPDATE] hint: among the three coding questions, p3 is the easiest, and p1 is similar to
p3.
         you'll realize that both are very similar to p0 (fibonacci).
         p2 is slightly different from these, but also very easy.

0. Is Fibonacci REALLY O(n)?
   Hint: f(n) itself grows exponentially.

1. [WILL BE GRADED]
   Maximum Weighted Independent Set

   [HINT] independent set is a set where no two numbers are neighbors in the original
list.
          see also https://en.wikipedia.org/wiki/Independent_set_(graph_theory)

   input:  a list of numbers (could be negative)
   output: a pair of the max sum and the list of numbers chosen

   >>> max_wis([7,8,5])
   (12, [7,5])

   >>> max_wis([-1,8,10])
   (10, [10])

   >>> max_wis([])
   (0, [])

   [HINT] if all numbers are negative, the optimal solution is 0,
          since [] is an independent set according to the definition above.

   >>> max_wis([-5, -1, -4])
   (0, [])

   What's the complexity?

   Include both top-down (max_wis()) and bottom-up (max_wis2()) solutions,
   and make sure they produce exact same results.
   We'll only grade the top-down version.

   Tie-breaking: any best solution is considered correct.

   Filename: mis.py

   [HINT] you can also use the naive O(2^n) exhaustive search method to verify your
answer.


2. Number of n-node BSTs

   input: n
   output: number of n-node BSTs

   >>> bsts(2)
```

```
    2
    >>> bsts(3)
    5
    >>> bsts(5)
    42

    [HINT] there are two 2-node BSTs:
       2    1
      /      \
     1        2
    Note that all other 2-node BSTs are *isomorphic* to either one.

    What's the complexity of this DP?

    What's the name of this famous number series?

    Feel free to use any implementation style.

    Filename: bsts.py

 3. Number of bit strings of length n that has

    1) no two consecutive 0s.
    2) two consecutive 0s.

    >>> num_no(3)
    5
    >>> num_yes(3)
    3

    [HINT] There are three 3-bit 0/1-strings that have two consecutive 0s.
             001  100  000
           The other five 3-bit 0/1-strings have no two consecutive 0s:
             010  011  101  110  111

    Feel free to choose any implementation style.

    Filename: bitstrings.py

    [HINT] like problem 1, you can also use the O(2^n) exhaustive search method to verify
 your answer.


 Debriefing (required!): --------------------------

 0. What's your name?
 1. Approximately how many hours did you spend on this assignment?
 2. Would you rate it as easy, moderate, or difficult?
 3. Did you work on it mostly alone, or mostly with other people?
 4. How deeply do you feel you understand the material it covers (0%-100%)?
 5. Which part(s) of the course you like the most so far?
 6. Which part(s) of the course you dislike the most so far?

 This section is intended to help us calibrate the homework assignments.
 Your answers to this section will *not* affect your grade; however, skipping it
 will certainly do.
```