1 Independent Set on Paths

Let $G = (V, E)$ be an undirected graph with n nodes. A subset of the nodes is called an **independent set** if no two of them are joined by an edge. Finding large independent sets is difficult in general; but here we'll see that it can be done efficiently if the graph is "simple" enough.

Call a graph $G = (V, E)$ a **path** if its nodes can be written as $v_1, v_2, \ldots, v_n$, with an edge between v_i and v_j if and only if the numbers i and j differ by exactly 1. With each node v_i, we associate a positive integer *weight* w_i.

The goal in this exercise is to solve the following problem:

Find an independent set in a path G whose total weight is as large as possible.

- Draw the five-node path with node weights (in order): 1, 8, 6, 3, 6.
- Give an example to show that the following algorithm *does not* always find an independent set of maximum total weight:

```
(The "heaviest-first" greedy algorithm)
  Start with S equal to the empty set
  While some node remains in G:
    Pick a node v_i of maximum weight
    Add v_i to S
    Delete v_i and its neighbors from G

  Return S
```

- Give an example to show that the following algorithm also *does not* always find an independent set of maximum total weight.

```
Let S_1 be the set of all v_i where i is an odd number.
Let S_2 be the set of all v_i where i is an even number.
(Note that both S_1 and S_2 are independent sets.)
Determine the greater of the sum of weights in S_1 or S_2
    and return that set.
```

- Describe an algorithm that takes an n-node path G with weights and returns an independent set of maximum total weight. The running time should be polynomial in n, independent of the values of the weights. (Try, if you can, to define and solve a recurrence relation for the running time of your algorithm.)

2 Magic Squares

You may use the Internet to search for references, explanations, proposed solutions to this exercise. Explain answers to the questions below in your own words.

A magic square of order n is an arrangement of the integers from 1 to n^2 in an $n \times n$ matrix, with each number occurring exactly once, so that each row, each column, and each main diagonal has the same sum.

- For a magic square of order n, what does the sum of each row equal? Prove that.
- Describe a backtracking algorithm (in the form of how backtracking algorithms are described in our textbook and how we've covered in class) to generate *a* magic square.

3 Addition Chain

Exercise 3(a), Chapter 2 (page 94) in the textbook.