This homework is due Friday March 6th at the start of class. Remember to justify your work even if the problem does not explicitly say so. Writing your solutions in \LaTeX is recommend though not required.

**Question 1** (Counting Integer Partitions, 30 points). Define a partition of an integer \( n \) to be a sequence of integers \( 0 < a_1 \leq a_2 \leq \ldots \leq a_k \) so that \( \sum_{i=1}^{k} a_k = n \). So, for example, the partitions of 5 are 

\[ 1 + 1 + 1 + 1 + 1, \ 1 + 1 + 1 + 2, \ 1 + 1 + 3, \ 1 + 2 + 2, \ 2 + 3, \ 5. \]

Come up with an algorithm that given \( n \) computes the number of partitions of \( n \) in time that is polynomial in \( n \). [10 points for algorithm, 10 points for runtime analysis, 10 points for correctness proof]

**Question 2** (Random Walks in Graphs, 30 points). Let \( G \) be a graph with a particular vertex \( s \). Consider the discrete time random walk on \( G \) starting at \( s \). Namely, a person starts at \( s \) at time \( t = 0 \). Then at each timestep, the person moves to a random adjacent vertex. So if they are at vertex \( v \) at time \( t \), then at time \( t + 1 \), they will move to a random adjacent vertex (each with equal probability).

Find an algorithm that given a graph \( G \), a vertex \( s \) and an integer \( t \) and another vertex \( v \) computes the probability that the random walk starting at \( s \) will be at vertex \( v \) at time \( t \). You algorithm should have runtime polynomial in \( t \) and the number of edges and vertices of \( G \). [10 points for algorithm, 10 points for runtime analysis, 10 points for correctness proof]

**Question 3** (Tour Scheduling, 40 points). Hal is planning a concert tour for his band. Several possible venues have given him offers to hold shows. The \( i^{th} \) venue is willing to pay \( R_i \) for his band but would keep them occupied for an interval of time \( I_i \). Hal knows that he cannot book his band at two venues that require them to be occupied for overlapping time intervals.

(a) Give an algorithm to find a schedule of venues to perform at that makes as much total money as possible without booking two venues for overlapping times. [10 points for algorithm, 10 points for runtime analysis, 10 points for correctness proof]

(b) Hal realizes that there is also a cost of travelling between venues. Suppose that for each pair of venues \( i \) and \( j \) there is a cost \( C(i,j) \) to travel from one location to the other. How would you need to modify the above algorithm to minimize the total revenue minus travel costs? [10 points for new algorithm]

The algorithms in each of these parts should be polynomial time in the number of venues.

**Question 4** (Extra credit, 1 point). Approximately how much time did you spend working on this homework?