**Algorithms**
**Dynamic Programming Group Project (75 pts)**

This project requires you to first theoretically solve the dynamic programming problem below and then write a program that implements your solution. [1] **You are not allowed to use the internet or consult any references. The only people you can work with on this project are your group members.**

1. *Problem Description*:

   The Astronomy club is faced with the following algorithmic problem. There are $n$ consecutive astronomical events they could observe on a particular night that occur exactly one minute apart. Thus event $j$ occurs at minute $j$. Also, event $j$ occurs at integer coordinate $d_j$ in the sky (we're assuming the sky is one-dimensional). The telescope's initial position at minute 0 is assumed to be coordinate 0 and the club is required to observe the last event $n$ (occurring at minute $n$). The catch here is that the telescope can only be moved one coordinate per minute. So, at minute 1, the telescope can be moved to coordinate location 1 or $-1$ (or it could remain at location 0).

   The optimization problem you have to solve is: given the coordinates of each of the $n$ events, find a viewable subset of *maximum* size, subject to the requirement that it should contain event $n$.

   **Example**: In the example below, the optimal solution is to observe events {1,3,6,9}. Note that the telescope has time to move from one event in this set to the next event moving at one coordinate location per minute:

| Event | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| Coordinate | 1 | $-4$ | $-1$ | 4 | 5 | $-4$ | 6 | 7 | $-2$ |

2. *Deliverables*: Please submit a **hard copy** of all of the items requested below. Please don't send this by email as I would have to print it out anyway.

   (a) [40 pts] Theory: Devise an efficient dynamic programming algorithm that finds an optimal solution. Demonstrate each of the following dynamic programming steps:

      i. Describe the main idea of your approach including how you break the problem into smaller recursive problems. A good way to do this is to develop notation and provide a recurrence relation [20 pts].

      ii. Write pseudocode for your dynamic programming algorithm [10 pts].

      iii. Develop a traceback algorithm that returns an optimal subset of viewable events [10 pts].

   (b) [10 pts] Theory: Derive the complexity of your algorithm.

   (c) [15 pts] Implementation: Implement your algorithm and submit the print-out of your code. If you were unable to get your code to compile/run, please state this clearly. Although we don't plan to run everybody's code, we might choose a few groups randomly and ask them to demonstrate that their code works. (It would look pretty bad if you claim your code runs, but it doesn't!)

   (d) [10 pts] Implementation: Demonstrate that your code works correctly by showing its results on a small example. (Provide the example you used so that we can verify this while grading.)

   (e) Submit group effort percentages and indicate who did what. These must be submitted individually by each group member. See syllabus to determine how I will use this to compute your individual grade.

[1] The problem has been adapted from the text by Kleinberg and Tardos