Introduction

- What are Carryover Effects?
- How does the program work?
  - Basic structure
  - Algorithms used - Code Specific
  - Final Result
- What were the results?
What are Carryover Effects?

- A Sample Set.
- 1-2  1-3  1-4  1-5  1-6  1-7  1-8
- 3-4  2-4  2-3  2-6  2-5  2-8  2-7
- 5-6  5-8  5-7  3-7  3-8  3-5  3-6
- 7-8  6-7  6-8  4-8  4-7  4-6  4-5
What are Carryover Effects?

- The algorithm used to calculate them.
  - Calculating a carryover.
    - 1-2 [1-3] ...
    - [3-4] 2-4 ...
    - ... ... ...
  - Detecting multiple occurrences.
    - 3 1 8
    - 2 4 5
    - ...
What are Carryover Effects?

- Sample Set revisited.
  - 1-2 [1-3] (1-4) 1-5 (1-6) 1-7 1-8
  - [3-4](2-4) 2-3 (2-6) 2-5 2-8 2-7
  - 5-6 5-8 5-7 3-7 3-8 3-5 3-6
  - 7-8 6-7 6-8 4-8 4-7 4-6 4-5
What are Carryover Effects?

- The conjecture behind them.
  - Only number of teams that are a power of 2

- What I am trying to achieve.
  - There is such a case with 12 teams.
How does the Program Work?

- Basic Structure
  - Core Algorithm
    - Works with any case
    - Receives input, computes carryover effects, prints output if it is successful.
  - Sorting Algorithm
    - Used to get all possible combinations of input.
How does the Program Work?

Algorithms used - Code Specific

- The Core Algorithm
  - Uses 2 dimensional arrays to hold the input Data and the carryover values.
  - Uses the algorithm defined earlier to calculate a carryover effect.
  - Checks for multiple occurrences while calculating.
  - Uses an efficient sort to keep from going through all cases
How does the Program Work?

Algorithms used - Code Specific

The Sorting Algorithm

• Hard coded to work with 12 cases only.
• Uses binary numbers to calculate a different data set.
• Example of a sort
  – 1-2  1-3  1-4  1-5  1-6  1-7  1-8  1-9  1-10 ...
  – 3-4  2-4  2-3  2-6  2-5  2-8  2-7  2-10  2-9 ...
  – 5-6  5-9  5-10  3-11  3-12  3-5  3-6  3-7  3-8 ...
  – 7-8  6-10  6-9  4-12  4-11  4-6  4-5  4-8  4-7 ...
  – ...  ...  ...  ...  ...  ...  ...  ...  ...

How does the Program Work?

- **Final Result**
  - Prints out a list of operations to a file.
  - If an answer was found it prints a successful message to the screen and writes the successful input along with its carryover effects to a file.
  - If no case is found, it simply prints a message saying that it did not find an answer.
What were the Results?

- **Program Output**
  - The program found the answer for 8 teams, which is a power of 2.
  - The program did not find an answer for 12 teams.

- **Conclusions**
  - The conjecture holds true for 8 and 12 teams.