Algorithms
Homework 3 Specification

Note

Skeleton code is available in four different programming languages: C, C++, Java, Python. You may choose any one of them to complete the assignment. If you wish to use a language other than the four provided, please contact the TA (`ta@theory.snu.ac.kr`).

N-Queens with Holes

Input format. The first line contains two integers: $4 \leq n \leq 10$ and h, where $h \leq 3$ is the number of holes on the board. The next h lines each contain two integers indicating the row and column position of a hole (0-based index).
e.g.

        5 1
        1 3

The coordinate system follows the standard matrix-style indexing: the top-left corner is $(0,0)$, with the first index representing the row and the second index representing the column. For example, in the board shown below, the hole marked with 'o' is at position $(1,3)$.

$$x \quad x \quad x \quad x \quad x$$
$$x \quad x \quad x \quad o \quad x$$
$$x \quad x \quad x \quad x \quad x$$
$$x \quad x \quad x \quad x \quad x$$
$$x \quad x \quad x \quad x \quad x$$

Implementation. The skeleton code already includes input parsing and timing functionality. You need to implement the following two functions in `src/hw3.xx`: `solve_iterative_nqueens` and `solve_recursive_nqueens`. Please refer to the skeleton code for the input and output specifications of each function. **There are no helper functions provided, but you are free to define your own and use any standard library functions or features.**

Compilation and Execution. For all languages, you can compile the code using `make`, and run the program using `make run` in command line. (For Python, compilation is not required.) If you want to check the specific commands used, please refer to the provided `Makefile` for each language.

Output. If all functions are correctly implemented according to the specifications, running the program should produce output similar to the following:

```
Iterative: 12345
Elapsed Time (ms): 0.274850
Recursive: 12345
Elapsed Time (ms): 0.000882
```

Grading

For each instance, your program may use up to 1 second for C/C++ (5 seconds for Python and Java) and 1 GB of memory.

Environment. We suggest using one of C, C++, Java or Python. If it is difficult for you to use one of the languages mentioned above, please contact the TA (`ta@theory.snu.ac.kr`). The compilers/interpreters and compilation commands that will be used for grading are as follows:

- C: gcc (Debian 12.2.0-14) 12.2.0
  `gcc -O2 -std=gnu99 -Iinclude src/main.c src/hw3.c -o main`

- C++: g++ (Debian 12.2.0-14) 12.2.0
  `g++ -O2 -std=gnu++17 -Iinclude src/main.cpp src/hw3.cpp -o main`

- Python: Python 3.11.5

- Java: javac (OpenJDK-23)
  `mkdir -p bin`
  `javac -d bin Main.java src/Hole.java src/HW3.java`

Submission.
Please compress your submission as `AG_HW3_[student ID].zip` (e.g., `AG_HW3_2025-12345.zip`). Your submission should include exactly 3 files:

- Source code for the problem (Choose one from `hw3.c`, `hw3.cpp`, `hw3.py`, `HW3.java`)

- Report (`report.pdf`)

- One additional example (`2.in`)

Hand in your submission to eTL. Please make sure that the submission files are placed at the top level of the compression and that submission file names are the same as above. **Please follow the rules for submission file names.**