Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное учреждение высшего образования

«Пермский национальный исследовательский политехнический университет»

ПНИПУ

Отчет по лабораторной работе

**«Задача о восьми ферзях»**

Выполнил:

Студент группы РИС-23-2б

Коротаев Антон Алексеевич

Проверила:

доцент кафедры ИТАС

О.А. Полякова

2024 г.

**Создание алгоритма для выполнения задачи**

**Постановка задачи:**

Расставить на стандартной 64-клеточной шахматной доске 8 ферзей так, чтобы ни один из них не находился под боем другого.

Подразумевается, что ферзь бьёт все клетки, расположенные по вертикалям, горизонталям и обеим диагоналям.

**Словесный алгоритм:**

1. **Инициализация доски**: Создается массив board[8][8], представляющий шахматную доску. Каждая клетка массива изначально содержит значение 0, что означает, что эта клетка не находится под боем ни от одного ферзя.
2. **Установка ферзя**: Для установки ферзя на определенную клетку доски вызывается функция setQueen(int i, int j). Эта функция инкрементирует значения во всех клетках, которые бьются этим ферзем. Это включает в себя клетки на вертикалях, горизонталях и обеих диагоналях относительно текущей клетки. Клетка, на которой ферзь находится, помечается значением -1.
3. **Отмена установки ферзя**: Если ферзь не может быть успешно размещен в текущей позиции, вызывается функция resetQueen(int i, int j), чтобы отменить установку ферзя. Эта функция декрементирует значения в клетках, которые ранее были увеличены при установке ферзя, и снимает пометку с текущей клетки.
4. **Попытка размещения ферзя в текущей строке**: Функция tryQueen(int i) рекурсивно вызывается для размещения ферзя в строке i. Она проходит по всем клеткам в этой строке и пытается установить ферзя в каждую клетку, если она не находится под боем. Если ферзь успешно установлен, функция вызывается для следующей строки. Если не удается установить ферзя в текущей строке, предыдущий ферзь удаляется, и пытается установить ферзя в другую клетку в текущей строке.
5. **Главная функция и вывод результата**: В главной функции main() происходит инициализация доски и вызов tryQueen(0) для начала попыток размещения ферзей. После успешного размещения всех ферзей на доске выводится итоговая расстановка ферзей.
6. Этот процесс продолжается рекурсивно, пока все ферзи не будут успешно размещены на доске или пока не будет перебрано все возможные комбинации.

**Описание работы каждой функции**

1. setQueen(int i, int j)

|  |
| --- |
| void setQueen(int i, int j)  {  for (int x = 0; x < 8; ++x)  {  ++board[x][j]; // Увеличиваем количество атакующих клеток по вертикали  ++board[i][x]; // Увеличиваем количество атакующих клеток по горизонтали  int foo;  foo = j - i + x;  if (foo >= 0 && foo < 8)  ++board[x][foo]; // Увеличиваем количество атакующих клеток по диагонали слева сверху направо вниз  foo = j + i - x;  if (foo >= 0 && foo < 8)  ++board[x][foo]; // Увеличиваем количество атакующих клеток по диагонали слева вниз направо вверх  }  board[i][j] = -1; // Устанавливаем ферзя на данную клетку  } |

Функция setQueen(int i, int j) используется для установки ферзя на конкретную клетку доски с координатами (i, j). Вот подробное описание того, как работает эта функция:

1. Инкрементация значений: Когда ферзь устанавливается на клетку с координатами (i, j), функция начинает инкрементировать значения во всех клетках, которые могут быть атакованы этим ферзем. Для этого происходит следующее:

* Каждая клетка в той же строке, где находится установленный ферзь, получает инкрементированное значение.
* Каждая клетка в том же столбце, где находится установленный ферзь, получает инкрементированное значение.
* Каждая клетка на диагонали, проходящей через установленный ферзь, получает инкрементированное значение.

1. Инкрементация по горизонтали и вертикали: Для каждой клетки в текущей строке и столбце, кроме той, где находится установленный ферзь, инкрементируется значение. Это означает, что все клетки в текущей строке и столбце, кроме клетки с установленным ферзем, теперь находятся под угрозой этого ферзя.
2. Инкрементация по диагоналям: Затем функция вычисляет, какие клетки находятся на диагоналях относительно установленного ферзя, и инкрементирует их значения. Это включает в себя клетки на диагонали слева сверху направо вниз и клетки на диагонали слева вниз направо вверх.
3. Отметка клетки с ферзем: Наконец, клетка, где установлен ферзь (координаты (i, j)), помечается значением -1, чтобы обозначить, что на этой клетке находится ферзь.

Этот процесс обеспечивает обновление состояния доски после установки ферзя, чтобы можно было корректно вычислить, какие клетки находятся под угрозой этого ферзя.

2. resetQueen(int i, int j)

|  |
| --- |
| void resetQueen(int i, int j)  {  for (int x = 0; x < 8; ++x)  {  --board[x][j]; // Уменьшаем количество атакующих клеток по вертикали  --board[i][x]; // Уменьшаем количество атакующих клеток по горизонтали  int foo;  foo = j - i + x;  if (foo >= 0 && foo < 8)  --board[x][foo]; // Уменьшаем количество атакующих клеток по диагонали слева сверху направо вниз  foo = j + i - x;  if (foo >= 0 && foo < 8)  --board[x][foo]; // Уменьшаем количество атакующих клеток по диагонали слева вниз направо вверх  }  board[i][j] = 0; // Убираем ферзя с данной клетки  } |

Функция resetQueen(int i, int j) используется для отмены установки ферзя на клетку с координатами (i, j). Вот подробное описание того, как работает эта функция:

1. Декрементация значений: Когда ферзь удаляется с клетки с координатами (i, j), функция начинает декрементировать значения во всех клетках, которые были изменены при установке этого ферзя. Для этого происходит следующее:

* Для каждой клетки в той же строке, где находился убранный ферзь, уменьшается значение.
* Для каждой клетки в том же столбце, где находился убранный ферзь, уменьшается значение.
* Для каждой клетки на диагонали, проходящей через убранный ферзь, уменьшается значение.

1. Декрементация по горизонтали и вертикали: Для каждой клетки в текущей строке и столбце, кроме той, где находился убранный ферзь, декрементируется значение. Это означает, что все клетки в текущей строке и столбце, кроме клетки, где находился убранный ферзь, больше не находятся под угрозой этого ферзя.
2. Декрементация по диагоналям: Затем функция вычисляет, какие клетки находились на диагоналях относительно убранного ферзя, и декрементирует их значения. Это включает в себя клетки на диагонали слева сверху направо вниз и клетки на диагонали слева вниз направо вверх.
3. Удаление пометки с клетки с ферзем: Наконец, клетка, где ранее находился ферзь (координаты (i, j)), снова помечается значением 0, чтобы обозначить, что на этой клетке больше нет ферзя.

Этот процесс обеспечивает корректное обновление состояния доски после удаления ферзя, чтобы корректно вычислить, какие клетки больше не находятся под угрозой этого ферзя.

3. tryQueen(int i)

|  |
| --- |
| bool tryQueen(int i)  {  bool result = false;  for (int j = 0; j < 8; ++j)  {  if (board[i][j] == 0) // Если клетка свободна от атакующих ферзей  {  setQueen(i, j); // Устанавливаем ферзя  if (i == 7) // Если это последняя строка (последний ферзь размещен успешно)  {  result = true; // Успех  }  else  {  if (!(result = tryQueen(i + 1))) // Рекурсивно пробуем разместить следующего ферзя  resetQueen(i, j); // Если не удалось, отменяем установку ферзя  }  }  if (result) // Если ферзь успешно установлен, выходим из цикла  break;  }  return result; // Возвращаем результат попытки установки ферзя  } |

Функция tryQueen(int i) является ключевой частью алгоритма и используется для попытки размещения ферзя в текущей строке доски. Вот подробное описание того, как работает эта функция:

1. Попытка размещения ферзя в текущей строке: Функция начинает свою работу с попытки установки ферзя в текущей строке i. Она перебирает все клетки в этой строке, поочередно пытаясь разместить ферзя в каждой из них.
2. Проверка возможности установки ферзя: Для каждой клетки в текущей строке функция проверяет, можно ли установить ферзя в эту клетку. Если значение в клетке равно 0, это означает, что данная клетка не находится под угрозой других ферзей и ферзь может быть установлен.
3. Установка ферзя: Если ферзь может быть установлен в текущую клетку, вызывается функция setQueen(i, j), где j - номер столбца, в котором будет размещен ферзь. Затем функция проверяет, является ли текущая строка последней строкой доски. Если это так, значит, все ферзи успешно размещены, и функция возвращает true.
4. Рекурсивный вызов: Если текущая строка не является последней, функция вызывает саму себя рекурсивно для размещения следующего ферзя в следующей строке i + 1.
5. Отмена установки ферзя: Если размещение следующего ферзя в следующей строке i + 1 не удается, функция вызывает resetQueen(i, j) для отмены установки ферзя в текущей клетке. Это позволяет перейти к следующей клетке в текущей строке и продолжить попытки размещения ферзя.
6. Возврат результата: Если ферзь успешно размещен в текущей строке, функция возвращает true, что указывает на успешное завершение размещения всех ферзей на доске. В противном случае функция возвращает false, что указывает на то, что текущий ферзь не может быть успешно размещен, и перебор должен продолжаться с другой клетки в текущей строке.

В функции main() происходит инициализация доски и запуск процесса размещения ферзей на шахматной доске. Вот подробное описание процессов, происходящих в функции main():

1. Инициализация доски:

* Создается двумерный массив board[8][8], представляющий шахматную доску. В каждой ячейке этого массива будет храниться число, отражающее количество ферзей, угрожающих данной клетке. Изначально все значения устанавливаются в 0.

1. Попытка размещения ферзей:

* Затем вызывается функция tryQueen(0), которая начинает процесс размещения ферзей. Первый ферзь будет пытаться разместиться в первой строке (с индексом 0).
* Функция tryQueen(0) будет рекурсивно вызывать себя для попыток размещения ферзей в каждой строке доски, начиная с первой.

1. Проверка успешного размещения ферзей:

* Если функция tryQueen(0) успешно размещает все 8 ферзей на доске, она вернет значение true, указывая на успешное завершение процесса размещения ферзей.
* В противном случае, если ферзи не могут быть размещены таким образом, чтобы ни один не угрожал другому, функция tryQueen(0) вернет false.

1. Вывод результата:

* После завершения процесса размещения ферзей на доске, результат будет выведен на экран.
* Вывод осуществляется с помощью двух вложенных циклов for, которые проходят по всем клеткам доски. Каждая клетка с ферзем помечается символом "[]", а пустая клетка - точкой ".".

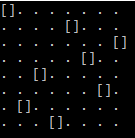
1. Завершение работы программы:

* После вывода результата программа завершает свою работу, возвращая значение 0.

**Код для решения задачи**

|  |
| --- |
| #include <iostream>  using namespace std;  int board[8][8];  void setQueen(int i, int j)  {  for (int x = 0; x < 8; ++x)  {  ++board[x][j];  ++board[i][x];  int foo;  foo = j - i + x;  if (foo >= 0 && foo < 8)  ++board[x][foo];  foo = j + i - x;  if (foo >= 0 && foo < 8)  ++board[x][foo];  }  board[i][j] = -1;  }  void resetQueen(int i, int j)  {  for (int x = 0; x < 8; ++x)  {  --board[x][j];  --board[i][x];  int foo;  foo = j - i + x;  if (foo >= 0 && foo < 8)  --board[x][foo];  foo = j + i - x;  if (foo >= 0 && foo < 8)  --board[x][foo];  }  board[i][j] = 0;  }  bool tryQueen(int i)  {  bool result = false;  for (int j = 0; j < 8; ++j)  {  if (board[i][j] == 0)  {  setQueen(i, j);  if (i == 7)  {  result = true;  }  else  {  if (!(result = tryQueen(i + 1)))  resetQueen(i, j);  }  }  if (result)  break;  }  return result;  }  int main()  {    for (int i = 0; i < 8; ++i)  {  for (int j = 0; j < 8; ++j)  {  board[i][j] = 0;  }  }  tryQueen(0);  for (int i = 0; i < 8; ++i)  {  for (int j = 0; j < 8; ++j)  {  if (board[i][j] == -1)  cout << "[]";  else  cout << ". ";  }  cout << endl;  }  return 0;  } |

**Работа программы**

****