



VI Поволжская олимпиада по информационным технологиям среди студентов и аспирантов «Волга ИТ – 2013»

Номинация «Прикладное программирование»

Задание отборочного этапа

Постановка задачи

Требуется создать приложение, которое бы помогло пользователю решать шахматные задачи – мат в N ($1 \leq N \leq 3$) ходов. Белые ходят первыми и выигрывают. Максимальное количество фигур с каждой стороны — 5. Необходимо найти и вывести все последовательности ходов, ведущих к мату черного короля.

Приложение должно загрузить данные из файла `input.txt`, вывести все найденные решения в файл `output.txt` не позднее чем через 3 секунды после запуска. Приложение должно иметь простой графический интерфейс для демонстрации найденных решений.

Требованию к графическому интерфейсу:

- Отображение шахматной доски
- Список всех найденных решений и возможность выбора одного из них
- Кнопку для проигрывания на шахматной доске следующего хода одной из сторон

Любые улучшения приветствуются.

Шахматные ситуации, которые не учитываются:

- Превращение пешек в другую фигуру
- Взятие пешки на проходе

Входные данные

Файл имеет кодировку ASCII.

Первая строка содержит N - максимальное количество ходов за которое решается задача.

Вторая строка содержит список белых фигур через пробел, где

- К — Король
- Q — Ферзь
- R — Ладья
- N — Конь
- В — Слон
- р (маленькая Р) — Пешка
- a-h — Позиция по горизонтали слева-направо
- 1-8 — Позиция по вертикали снизу-вверх

в формате:

<идентификатор фигуры><позиция по горизонтали><позиция по вертикали>

Третья строка содержит список черных фигур в том же формате.

Пример:

2

Kc1 Nf3

Ka1 pa2 pa6

Выходные данные

Файл должен иметь кодировку ASCII.

Первая строка должна содержать количество найденных комбинаций.

Следующие строки содержат информацию о каждой из комбинаций в формате:

<Количество ходов до мата>

<Первый ход белых> <Первый ход черных>

<Второй ход белых> <Второй ход черных>

...

по одному ходу на каждой строке, где каждый ход имеет вид:

<Фигура><начальная позиция>-<конечная позиция>

В случае, если лимит времени исчерпан алгоритм должен прекратить свое выполнение и вывести в выходной файл все решения, которые были найдены за отведенное время. Если было не найдено ни одного решения, выходной файл должен содержать единственное число 0.

Разрешается ставить лишние пробелы в конце строк и пустые строки в конце файла.

Пример:

2

2

Nf3-d2 pa6-a5

Nd2-b3

2

Nf3-d4 pa6-a5

Nd4-b3

Дополнительные условия

Программа предоставляется в виде исполняемого файла, исходных текстов, файлов проекта для среды разработки и readme.txt.

Рекомендуемые среды разработки:

- C++ : Qt Creator 2.8.1
- C# : Visual Studio 2010/2012 Express
- Java: Eclipse или IntelliJ IDEA

Постарайтесь свести к минимуму все внешние зависимости и использовать только инструменты, предоставляемые Java 7, .NET 4 и Qt 5.1.

В файл readme.txt включите

- Краткое описание алгоритма
- Инструкцию по сборке программы из исходных текстов (дополнительные условия, настройки среды - постарайтесь свести их к минимуму)
- Требования для запуска исполняемого файла (если не запустится, мы попробуем собрать заново из исходных текстов, но это минус)

Принимаются частичные решения: GUI реализован частично или отсутствует, поддержка только некоторых фигур, малую глубину поиска (например, только для $N=1$), и пр. В таком случае в файле readme.txt необходимо указать какие ограничения наложены на программу.

Оценка

Итоговая оценка будет складываться из результатов прогона на тестовых задачах, результатов ручного тестирования GUI и, возможно, анализа исходных текстов программы.

Вопросы и уточнения направляйте в группу <http://vk.com/volgait>.