# Алгоритмы

Шашки – игра с конечным числом состояний. Это означает, что, обладая бесконечными вычислительными ресурсами, мы смогли бы найти решение этой игры – то есть такую стратегию, следуя которой всегда можно было бы выиграть или, по крайней мере, сыграть вничью. Каждая позиция pos в шашках - это либо победа белых, либо победа черных, либо ничья. Определим функцию ***f:P→Z***, которая ставит в соответствие позиции ***pos*** из множества всевозможных позиций ***P*** целое число, отражающее «выгодность» этой позиции для текущего игрока***.*** Такая функция называется *оценочной функцией.*Если бы у нас был бесконечно быстрый компьютер, мы могли бы вычислить ее следующим образом:

* присвоим всем финальным позициям значения –1, 0, 1, в зависимости от исхода игры;
* применим рекурсивное правило**f**

где **pos** соответствует ходу текущего игрока, позиция **pos‘** соответствует ходу противника, а  **pos→pos‘** обозначает все допустимые ходы из позиции **pos**.

Чем больше значение функции, тем выгодней позиция. Таким образом, ставя каждому ходу в соответствие его оценку, мы бы могли однозначно определить лучший ход.

Количество [возможных позиций](http://en.wikipedia.org/wiki/Shannon_number) в шашках составляет около 5\*1020. Вычисления такого масштаба выполнить практически невозможно (канадским ученым потребовалось 200 компьютеров и 20 лет вычислений). Следовательно, находить наилучший ход нужно другим способом.

### Алгоритм полного перебора

Самым простым способом получить если не лучший, то хотя бы осмысленный ход, является метод полного перебора с возвратом в соединении с рекурсией. Если сгенерировать все возможные ходы из данной позиции для белых, затем для каждого такого хода повторить те же действия для черных, потом снова для белых и т.д., получится простой и понятный рекурсивный алгоритм, основанный на переборе позиций. Люди во время игры мыслят подобным образом. Однако, как уже было сказано, рассмотреть все возможные варианты до конца невозможно, поэтому приходится ограничивать глубину рекурсии. Пусть на каждом шаге известна текущая глубина рекурсии – параметр ***depth***, первоначально равный максимальной заданной глубине. Тогда если ***depth***≠0, то рекурсия будет вызываться с параметром ***depth-1***, в противном случае вычисления следует прекратить.

Блок-схема алгоритма полного перебора представлена на рис.1.

### Alpha-Beta с отсечениями

### Форсированные варианты

### Оценочная функция