

Оргмоменты

Одна контрольная — контекст на реализацию какого-то алгоритма. Пользоваться своим кодом запрещено.

Домашние задания представляют собой следующее: необходимо привести алгоритм, провести его теоретический анализ (доказать корректность и оценить время работы) и запрограммировать. Сначала сдаётся теория, потом практика.

Экзамен устный.

$$O_{\text{итоговая}} = 0.7 \cdot O_{\text{накопленная}} + 0.3 \cdot O_{\text{экзамен}}$$

$$O_{\text{накопленная}} = 0.2 \cdot O_{\text{КР}} + 0.12 \sum_{i=1}^5 O_{\text{ДЗ } i} + 0.2 \cdot O_{\text{семинары}}$$

Списывание, как обычно, *не поощряется*. С обсуждением ДЗ ситуация аналогична.

[Здесь](#) можно найти ссылки на ДЗ и краткое содержание лекций.

Автоматов *пока* не предусмотрено.

Литература:

- Кормен Т., Лейзерсон Ч., Ривест Р., Штайн К. — «Алгоритмы. Построение и анализ»
- Дасгупта С., Пападимитриу Х., Вазирани У. — «Алгоритмы»

Лекция 1 от 12.01.2016

Задача о Ханойских башнях. Три стержня.

Рассмотрим классическую задачу, предложенную Эдуардом Люка в 1883 году. Есть три стержня, при этом на первый стержень нанизано 8 дисков. Нужно перенести все диски на другой стержень, соблюдая два правила: диски можно двигать только по одному и нельзя класть диск большего радиуса на диск меньшего радиуса. Возможно ли это?

Придуманная профессором Люка легенда гласит, что в Великом храме города Бенарес, под собором, отмечающим середину мира, находится бронзовый диск, на котором укреплены 3 алмазных стержня, высотой в один локоть и толщиной с пчелу. Давным-давно, в самом начале времён, монахи этого монастыря провинились перед богом Брахмой. Разгневанный Брахма воздвиг три высоких стержня и на один из них возложил 64 диска, сделанных из чистого золота. Причем так, что каждый меньший диск лежит на большем.

Как только все 64 диска будут переложены со стержня, на который Брахма сложил их при создании мира, на другой стержень, башня вместе с храмом обратятся в пыль и под громовые раскаты погибнет мир.

Интуиция подсказывает, что это возможно. Но каков тогда алгоритм и сколько операций ему необходимо?

Пусть T_n — минимальное количество операций, за которое можно перенести n дисков. Сразу же заметим, что для переноса 0 дисков действий вообще не нужно. Тогда $T_0 = 0$.

Эксперименты с тремя дисками показывают, что наиболее оптимальный алгоритм состоит в следующем: перенести два диска на второй стержень, перенести третий диск на третий стержень, после чего перенести два диска со второго стержня на третий. Достаточно очевидно, что такое рассуждение работает для любого числа дисков. Отсюда можно сделать вывод, что

$$T_n \leq 2T_{n-1} + 1 \text{ для } n > 0.$$

Возникает логичный вопрос: а можно ли быстрее? Увы, но нет. Рано или поздно придётся перенести самый широкий диск. Но для этого необходимо поставить $n - 1$ диск на один стержень. Тогда можно сделать вывод, что

$$T_n \geq 2T_{n-1} + 1 \text{ для } n > 0.$$

Отсюда получаем, что минимальное количество операций задаётся следующим *рекуррентным соотношением*:

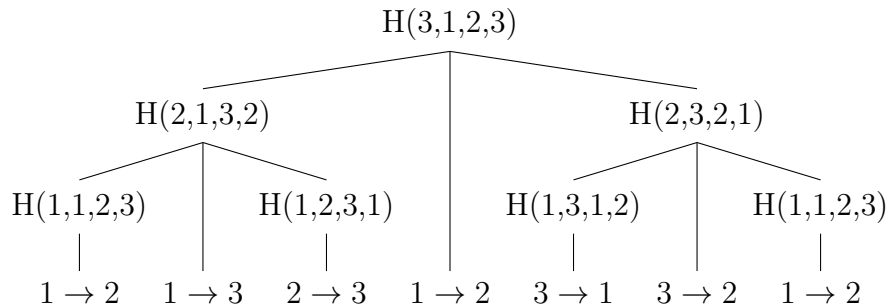
$$T_n = \begin{cases} 2T_{n-1} + 1, & n > 0 \\ 0, & n = 0 \end{cases}$$

Псевдокод для оптимального алгоритма перестановки дисков будет выглядеть так:

Algorithm 1 Рекурсивный алгоритм решения задачи о Ханойской башне

1: function HANOI3(n, i, j, k)	▷ n — количество дисков, i, j, k — номера стержней
2: if $n > 0$ then	▷ если есть, что переставлять
3: HANOI3($n - 1, i, j, k$)	▷ перенесём $n - 1$ диск со стержня i на стержень j
4: move $i \rightarrow k$	▷ переставим верхний диск со стержня i на стержень k
5: HANOI3($n - 1, k, j, i$)	▷ перенесём $n - 1$ диск со стержня j на стержень k

Покажем с помощью дерева операций, как алгоритм работает для трёх дисков:



По сути, данный алгоритм обходит дерево в глубину, выполняя необходимые перестановки.

Прекрасно. Мы знаем, что этот алгоритм наиболее оптимален и знаем рекуррентное соотношение, задающее количество операций. Но можно ли найти *замкнутую* формулу — такую, что она сразу даст нужное значение? Да, можно.

Свойство. $T_n = 2^n - 1$

Доказательство. Докажем это по индукции. База верна, так как $T_0 = 0 = 2^0 - 1$. Теперь пусть предположение верно для $n - 1$, то есть $T_{n-1} = 2^{n-1} - 1$. Тогда

$$T_n = 2T_{n-1} + 1 = 2(2^{n-1} - 1) + 1 = 2^n - 2 + 1 = 2^n - 1 \quad \square$$

Теперь немного изменим условие задачи.

Задача о Ханойской башне на четырёх стержнях

Условия те же, что и в стандартной задаче, но добавляется ещё один стержень. Можно ли сказать, что задача стала проще? Сложнее стать она точно не могла — никто не мешает игнорировать новый стержень.

Попробуем следующий алгоритм:

- Перенесём $n - 2$ диска с первого стержня на второй.

- Перенесём два диска с первого стержня на четвёртый, используя третий в качестве временного.
- Перенесём $n - 2$ диска со второго стержня на четвёртый.

Псевдокод для данного алгоритма будет иметь следующий вид:

Algorithm 2 Рекурсивный алгоритм решения задачи о Ханойской башне на 4-х стержнях, версия 1

```

1: function HANOI4( $n, i, j, k, l$ )           ▷  $n$  — количество дисков,  $i, j, k, l$  — номера стержней
2:   if  $n > 0$  then
3:     HANOI4( $n - 1, i, l, k, j$ )
4:     move  $i \rightarrow k$ 
5:     move  $i \rightarrow j$ 
6:     move  $k \rightarrow j$ 
7:     HANOI4( $n - 1, l, j, i, k$ )

```

Количество операций для данного алгоритма задаётся следующим рекуррентным соотношением:

$$T_n = \begin{cases} 2T_{n-2} + 3 & n > 1 \\ 1 & n = 1 \\ 0 & n = 0 \end{cases}$$

Достаточно логично, что T_n будет иметь вид $a \cdot 2^{\lfloor \frac{n}{2} \rfloor} + c$. В обозначениях, которые мы введём немного дальше, это будет иметь вид $\Theta(2^{\lfloor \frac{n}{2} \rfloor})$. Грубо говоря, это означает, что функция растёт примерно так же.

Неплохо. Но можно лучше? Да.

Рассмотрим принципиально другой алгоритм. Пусть количество дисков n имеет вид $n = \frac{m(m+1)}{2}$ для какого-то m . Тогда алгоритм будет иметь следующий вид:

Algorithm 3 Рекурсивный алгоритм решения задачи о Ханойской башне на 4-х стержнях, версия 2

```

1: function HANOI4( $n, i, j, k, l$ )           ▷  $n$  — количество дисков,  $i, j, k, l$  — номера стержней
2:   if  $n > 0$  then
3:     HANOI4( $n - m, i, l, k, j$ )
4:     HANOI3( $m, i, j, k$ )
5:     HANOI4( $n - m, l, j, i, k$ )

```

Если же $n = \frac{m(m+1)}{2} + c$, где $c \leq m$, то в первом шаге этого алгоритма вызывается тело функции с HANOI3(c, i, j, k).

Идейно данный алгоритм похож на то, что было предложено ранее. Но сколько операций он требует?

Построим дерево операций. Достаточно логично, что это будет полным бинарным деревом. Покажем, что в нём m уровней. Действительно, если $n_m = \frac{m(m+1)}{2}$, то $n_m - m = \frac{m(m+1)}{2} - m = \frac{m(m-1)}{2} = n_{m-1}$. На каждом уровне выполняется $2^i(2^{m-i} - 1) = 2^m - 2^i$. Тогда всего операций будет

$$\sum_{i=0}^m (2^m - 2^i) = \sum_{i=0}^m 2^m - \sum_{i=0}^m 2^i$$

Заметим, что $1 + 1 + 2 + 4 + \dots + 2^m = 2 + 2 + 4 + \dots + 2^m = 4 + 4 + \dots + 2^m = \dots = 2^{m+1}$. Тогда

$$\sum_{i=0}^m 2^m - \sum_{i=0}^m 2^i = (m+1) \cdot 2^m - 2^{m+1} + 1 = (m-1)2^m + 1$$

Теперь оценим время работы на большом количестве дисков. Так как $n = \frac{m(m+1)}{2} + c$ и $n \gg c$, то можно сказать, что $m \approx \sqrt{2n}$. Тогда

$$T_n \approx (\sqrt{2n} - 1)2^{\sqrt{2n}} + 1$$

Если воспользоваться обозначением Θ , то $T_n = \Theta(\sqrt{n}2^{\sqrt{2n}})$.

Данный алгоритм носит название *алгоритма Фрейма-Стюарта*. Есть гипотеза (названная в честь авторов этого алгоритма), что он является оптимальным (то есть нет алгоритма, работающего быстрее). Но пока что она не доказана.

Дальше пойдут теоретические изыски. Попробуем обобщить этот алгоритм для любого числа стержней:

Algorithm 4 Рекурсивный алгоритм решения задачи о Ханойской башне, общий случай

```

1: function HANOI( $n, i, j, P$ )                                 $\triangleright n$  — количество дисков,  $i, j$ , — основные стержни
2:                                                          $\triangleright P$  — множество вспомогательных стержней
3:   if  $n > 0$  then
4:     choose  $p \in P$ 
5:      $R := P \setminus p$ 
6:     if  $R = \emptyset$  then
7:       HANOI3( $n, i, j, p$ )
8:     else
9:       HANOI( $n - m, i, p, R \cup \{j\}$ )
10:      HANOI( $m, i, j, R$ )
11:      HANOI( $n - m, p, j, R \cup \{i\}$ )

```

Пусть для перемещения n дисков с помощью алгоритма на k стержнях ($k \geq 3$) нужно $h(n, k)$ операций. Тогда верно следующее:

$$h(n, k) = \begin{cases} 0 & n = 0 \\ 2^{n_m} - 1 & n > 0, k = 3 \\ 2h(n_{m-1}, k) + h(m, k - 1) & n > 0, k > 3 \end{cases}$$