Лекция по АиСД; 12.01.15

Попов Никита

12 января 2016 г.

Оргмоменты

Одна контрольная — запрограммировать некоторый алгоритм; либо пройденный (пользоваться ничем нельзя), либо

Задача — привести алгоритм, провести теоретический анализ и запрограммировать. Сначала сдаётся теория, потом практика.

Накопленная — $0.2 \times O_{\text{контрольная}} + 0.12 \times 5 \times O_{\text{домашние}} + 0.2 \times O_{\text{семинары}}$

Итоговая — $0.7 \times O_{\text{накопленая}} + 0.3 \times O_{\text{экзамен}}$

По списыванию всё как обычно, пользоваться своим кодом на контрольных нельзя, ДЗ предполагается не обсуждать.

wiki.cs.hse.ru — страница по курсу; ссылки на ДЗ, краткое содержание лекций.

Автоматов *пока* не предусмотрено.

Литература:

- Кормен, Лейзерсон, Ривест — “Алгоритмы. Построение и анализ”
- Дасгупта, Пападимитриу, Вазирани — “Алгоритмы”

Лекция

Ханойские башни

Три стержня; на первый стержне нанизаны 64 диска, от самого большого к самому маленькому. Задача: переложить все диски на второй стержень. Ограничения:

- Диски можно переносить только по одному;
- Диск нельзя класть на диск меньшего диаметра.

Какой может быть алгоритм? Варианты из аудитории:

1. Полный перебор
2. Рекурсивный алгоритм.

Рассмотрим такой рекурсивный алгоритм:

1. Переложим все, кроме n-ного, на третий стержень;
2. Переложим n-ный на второй;
3. Переложим все с третьего на второй стержень.

Запишем такой алгоритм:

```
Hanoi3(n,i,j,k)
    if n>0 then
        Hanoi3(n-1, i, k, j)
        move i \to j
        Hanoi3(n-1, k, j, i)
```

Дерево — см. тетрадь!

вставить картинку алгоритма и его работы

Алогритм, по сути, обходит это дерево в глубину и при этом слева направо, выполняя все перемещения, что встретятся.

Xbckj ifujd — n. Можно дерево рассмотреть как двоичное, если перемещения учитывать не в отдельных листьях, а в родительских узлах. В каждом узле, значит, мы выполняем одно действие, а в полном бинарном дереве $2^n - 1$ узел.

Докажем по индукции:

Пусть число перемещений для n дисков — $f(n) = \begin{cases} 0, & n = 0 \\ 2f(n-1) + 1, & n > 0 \end{cases}$

База: $f(0) = 2^0 = 1$ Переход: $f(n-1) = 2^{n-1} - 1; f(n) = 2(2^{n-1} - 1) + 1 = 2^n - 2 + 1 = 2^n - 1$

Но это время для этого алгоритма. А можно ли быстрее?

Рассмотрим некоторый алгоритм. Он рано или поздно должен переложить наибольший диск на второй стержень. Для этого ничего не должно быть на нём и на втором, т.е. все на третьем. А как получить эту конфигурацию? Оптимальным алгоритмом на $n-1$ шаг, что приводит к нашим вычислениям и уже полученному минимальному результату в $2^n - 1$.

Утверждение: задачу о Ханойских башнях нельзя решить за меньшее число шагов, причём решение с таким числом шагов ровно одно.

Доказательство по индукции:

База: $n = 0$: очевидно, решить быстрее, чем за 0 шагов нельзя и последовательность такая ровно одна.

Переход: предположим, что мы доказали это утверждение для $n-1$. Рассмотрим n: Рано или поздно алгоритму понадобится освободить первые два стержня, чтобы переложить первый диск на второй стержень. Необходимо сделать это одно перекладывание и после вернуть все оптимальным алгоритмом. Итого, опираясь на предположение индукции шагов в оптимальном и единственном решении для $n-1$, нам понадобится $2\left(2^{n-1} - 1\right) + 1$ шаг, что и равно $2^n - 1$.

Изменим задачу:

Четыре стержня

Условие в остальном ровно то же. Стала ли задача проще?

Сложнее она точно не стала, т.к. четвёртым можно просто не пользоваться.

Рассмотрев переход от двух к трём, кажется, что должно быть проще; как можно воспользоваться четвёртым?

Предложения:

- Переложить предпоследний отдельно на четвёртый и сэкономить на перекладывании башни из $n-1$, перекладывая вместо неё башню из $n-2$

```
Hanoi4(n,i,j,k,l)
    if n>0 then
        Hanoi4(n-2, i, k, j, l)
        move i to l
        move i to j
        move l to j
        Hanoi4(n-2, k, j, i, l)
```

Построив то же дерево, получим что в каждом узле три перемещения, а узлов $2^{\frac{n}{2}} - 1$ — экономия, но не очень большая.

Построим другой алгоритм:

```
Hanoi4(n,i,j,k,l)
    if n>0 then
        Hanoi4(n-m, i, l, j, k)
        Hanoi3(n, i, j, k)
        Hanoi4(n-m, l, j, i, k)
```

Заметим, что число шагов зависит от m. Пусть $n = \frac{m(m+1)}{2}$ (если n другое, то на первом шаге выберем такой m, чтобы $n-m$ было таким, а дальше на вход будет поступать число такого вида)

Получим то же двоичное дерево, но заметим, что число уровней в нём — m, т.к. на каждом шаге m уменьшается на 1 (ПОЧЕМУ? пояснить!), а в каждом узле работы проводится $2^m - 1$, т.к. мы пользуемся уже доказанным алгоритмом для трёх стержней.

Сложим по уровням. На i-ом уровне сумма всех узлов — $2^m - 2^i$, при нумерации с нуля.

$g(n) = \begin{cases} 0, m = 0 \\ g(n_{m-1}) + 2^m = -1, m > 0 \end{cases}$

$g(n) = \sum\limits_{i=0}^{m-1} (2^m - 2^i) = m * 2^m - \sum\limits_{i=0}^{m-1} 2^i = m * 2^m - (2^m - 1) = (m-1) \cdot 2^m + 1$

Докажем по индукции:

База: $m = 0; g(n_0) = 0 = -1 \cdot 2^0 + 1$

Переход: $g(m-1) = (m-2) \cdot 2^{m-1} + 1$ $g(m) = 2((m-2) \cdot 2^{m-1} + 1)$!!!!! СПИСАТЬ

$m \approx \sqrt{2n}; g(n) \approx \sqrt{2n} \cdot 2^{\sqrt{2n}}$

$g(n) = \Theta(\sqrt{2n} \cdot 2^{\sqrt{2n}})$

Попробуем обобщить этот алгоритм для любого числа стержней:

```
Hanoi(n, i,j,P)
    if n>0 then
        choose p \in P
        R := P\\\{p\}
        if R\neq\varnothong then
            Hanoi3(n, i, j, p)
        else
            Hanoi(n-m, i, p, R\cup\{j\})
            Hanoi(m, i, j, k)
            Hanoi(n-m, )<++>
```

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