Алгоритмы и структуры данных-1

Асимптотический анализ и рекуррентные соотношения. Часть 3

Практическое занятие 5 – 30.09-05.10.2024

2024-2025 учебный год

Методы решения рекуррентных соотношений

Метод подстановки

Используя метод подстановки, доказать или опровергнуть следующие утверждения об оценке временной сложности алгоритмов:

1. $T(n) = T(n-1) + 2 = \Theta(n)$
2. $T(n) = T(\lfloor n/2 \rfloor + 17) + n = O(n \log n)$

Дерево рекурсии

С помощью дерева рекурсии найти как можно более точную оценку временной сложности алгоритмов, время работы которых описывается следующими рекуррентными соотношениями:

1. $T(n) = 3T(^n/_2) + n$
2. $T(n) = T(n - a) + T(a) + cn$, где $a \geq 0$ и $c > 0$

Выполнить проверку с помощью метода подстановки.

Основная теорема

С помощью master-теоремы вычислить верхнюю оценку временной сложности алгоритмов со следующими рекуррентными соотношениями:

1. $T(n) = 16\,T(^n/_4) + n$
2. $T(n) = 4\,T(^n/_2) + n^2$
3. $T(n) = 2^n\,T(^n/_2) + n^n$
4. $T(n) = 2\,T(^n/_2) + n\log n$
5. $T(n) = 2\,T(^n/_2) + {}^n/_{\log n}$

Если master-теорема не может быть применена, дать необходимые пояснения.

Практические аспекты

Явная (closed) форма $T(n)$

$$\frac{1}{1\cdot 2}+\frac{1}{2\cdot 3}+\dots+\frac{1}{n(n+1)}=\frac{n}{n+1}$$

Вывести явную форму рекуррентного соотношения временной сложности $T(n)$ рекурсивного алгоритма A обработки некоторого массива размера n. Точное время работы A выражается следующим образом и зависит от среднего:

$$T(n)=\frac{2}{n}\big(T(0)+\dots+T(n-1)\big)+c.$$

Перемешивание части массива третями

```
shuffle(A, l, r)
if l < r
    thrd = (r − l) / 3
    shuffle(A, l, l + thrd − 1)
    shuffle(A, r − thrd + 1, r)
    for i = 0 to thrd − 1
        swap(A[l + thrd + i],
             A[l + 2 * thrd + i)
```

Перемешивание части массива третями

```
shuffle(A, l, r)
if l < r
    thrd = (r − l) / 3
    shuffle(A, l, l + thrd − 1)
    shuffle(A, r − thrd + 1, r)
    for i = 0 to thrd − 1
        swap(A[l + thrd + i],
             A[l + 2 * thrd + i)
```

1. Составить рекуррентное соотношение, которое описывает время работы алгоритма **shuffle**.
2. Вычислить верхнюю границу временной сложности данного алгоритма.

Домашнее задание

Задача трех T

Дано рекуррентное соотношение, которое описывает сложность некоторого алгоритма:

$$T(n) = \begin{cases} 2 \cdot T\left({}^{n}/_{2}\right) + n & n \mid 2 \\ 2 \cdot T(n-1) + n & n \nmid 2 \\ 1 & n = 1 \end{cases}$$

Попробуйте исследовать свойства дерева рекурсии (высоту, общее количество задач) и обосновать асимптотическую верхнюю границу временной сложности.