(3)

Ретроспекция

Анализ на итер. алг.

- Сумиране

$$\sum_{i=1}^{n} i^a \simeq n^{a+1} \qquad \sum_{\substack{i=n \\ i=2}}^{1} \frac{n}{i} = 2n-1$$

$\Theta(n)$

- Анализ на рек. алг. за търсене
- Анализ на алг. за сортиране

Инверсия

$arr[1 \ldots n] \quad i, j \in \{1 \ldots n\}$

$i < j \wedge arr[i] > arr[j]$

Алг за сортиране

- Времева сложност
- Пространствена сложност
- Устойчивост / стабилност

Метод на мехурчето

Прост. Сложност $\theta(1)$

- Устойчивост

сортиране

arr[1...n] ⟶

$T[i]$ - индексът на i-тия елемент от arr в сортирания масив.

arr = [5, 6, 2] → [2, 5, 6]

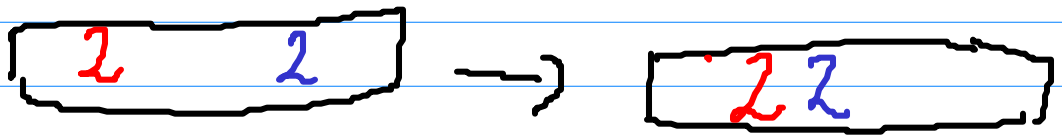
$T[1] = 2$ $\quad T[3] = 1$

$T[2] = 3$

Ще кажем, че алгоритъм за сортиране е устойчив, ако

$\forall i \forall j \in \{1...n\} \quad i < j \wedge arr[i] = arr[j]$

$\Rightarrow T[i] < T[j]$



Зависва относителната наредба

Пример

Студенти в курса по ДЗА

име, ФН, група

1) Сортиране по Ф.Н.

2) устойчиво сортиране по група

име, ФН, група

сортирани по Ф.Н

1
1
1

сортирани по Ф.Н

2
⋮
2

сортирани по Ф.Н.

3
⋮
3

? Методът НА мехурчето устойчив ли е?

ДА!

... 2 2 ...

swap? arr[i] > arr[i+1] ?

· Грешка

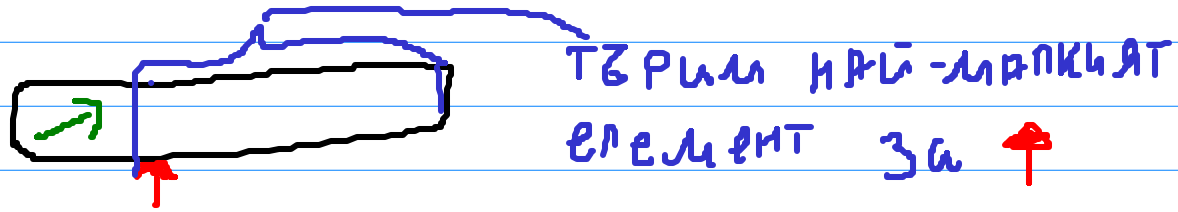
Сортиране с пряка селекция (Selection Sort)

Анализ:

1) Време сложност: $\Theta(n^2)$

2) Допълн. памет: $\Theta(1)$

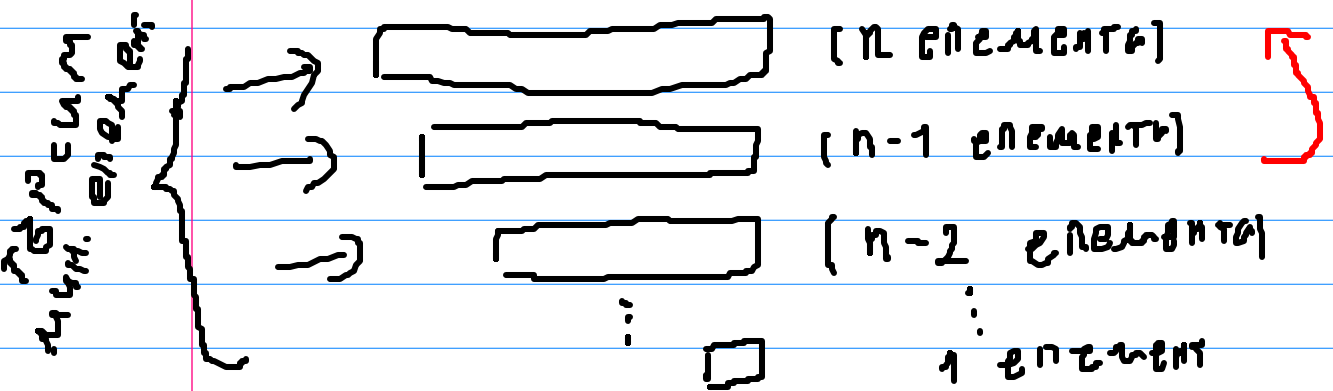
3) Устойчивост: Не!



1) Най-добър: $\Theta(n^2)$

Среден сл.: $\Theta(n^2)$

Най-лош сл.: $\sum_{i=1}^{n} i = \frac{n(n+1)}{2}$ $\Theta(n^2)$



2) Пространствена сложност

$\Theta(1)$

3) Устойчивост: Не

3 2 3 1 → 1 2 3 3

Предимство?

(!) • Минимален брой размени/swap-ове

Недостатъци: Анализ на сложността

сортиране чрез вмъкване

(Insertion Sort)

Анализът е подобен на

анализа на метода на мехурчето.

Зад 1

Каква е времевата сложност на следния алгоритъм

$$n \to \frac{n}{2} \to \frac{n}{4} \to \dots \to 1$$

$\log(n)$

$\Theta(\log(n))$

$n \& (n-1) == 0$

Ако n е степен на 2

$$\begin{array}{r} 10000000_2 \\ \& \quad 11111111 \\ \hline 00000000 \end{array}$$

Зад 3 Каква е сложността на алгоритъма F

$f(n)$

$g(1)\ g(2)\ g(3)\ \ldots\ g(n)$

g — наи-добрая сл. $\Theta(1)$

наи-лошая сл $\Theta(n)$

f наи-лошая сл

$n * \Theta(n) = \Theta(n^2)$ Не!

Амортизированный анализ

n увеличений на $g_{n-\log(n)}$

- "дешёвых" $\Theta(1)$
- "тяжёлых" $\Theta(n)$ $\log(n)$

$n \geq k$ =

$\sum_{i=1}^{n} 1$ — i не степень числа 2

$+ \sum_{i=1}^{n} i$ — i степень двойки

$=$

n изчисления на g

$$k + \sum_{i=1}^{n} i = \quad i \text{ е степен на } 2$$

$$= \underbrace{k}_{\leq n} + \underbrace{1+2+4+8 \dots \frac{n}{4} + \frac{n}{2} + n}_{2n-1} \leq$$

$$\leq 3n - 1$$

Копие на стрък едно изчисление на g!

$$\frac{3n-1}{n} \leq \boxed{3}$$

$\exists$ константа

$\Rightarrow$ Сложността на f:

$$\sum_{i=1}^{n} \theta(1) = \boxed{\theta(n)}$$

- Подобно на анализа на добавяне на елемент във вектор

- Пример за амортизирана сложност

  За добавяне на елемент в черно-червено дърво

  ⋮

Анализ на рекурсивни алгоритми

- решаване на рекурентни ур-я

  $a_0\ a_1 \cdot a_2 -- a_n$

  $a_0 = 1$

  $a_1 = 2$

  $a_n = a_{n-1} + 3a_{n-2}$

- Решаване на рек. ур-я

  1) Напукване + доказване

  2) Развиване

  3) Метода с характеристичното ур-е

  4) Master th.

5) Други методи

Решаване на рек. ур-е

Адмирале на общия ч-ло

за n-ия член. (без рекурсия)

$a_n = 2^n + 9^n + n^2$ ← пример

!

При анализа на рек. алгоритъм не ни интересува точното решение на ур-то, а неговата асимптотика

Задача Сложност на алгоритъма?

1) Съставяме рек ур-е

$$T(n) = T(n-1) + T(n-2) + 1$$

2) Решаване на

$$\underbrace{T(n) - T(n-1) - T(n-2)}_{} = 1$$

$\lambda^2 - \lambda - 1$ $\quad P(n) \cdot \alpha^n$

$n^0 \cdot 1^n$

↓

$\lambda_1 = \frac{1-\sqrt{5}}{2}$ $\lambda_2 = \frac{1+\sqrt{5}}{2}$

$C \cdot \eta^k \alpha^n$

k – степен

$\left\{ \frac{1-\sqrt{5}}{2}, \frac{1+\sqrt{5}}{2} \right\}_m$

Обхождаме в множеството $k+1$ пъти!

$\cup$

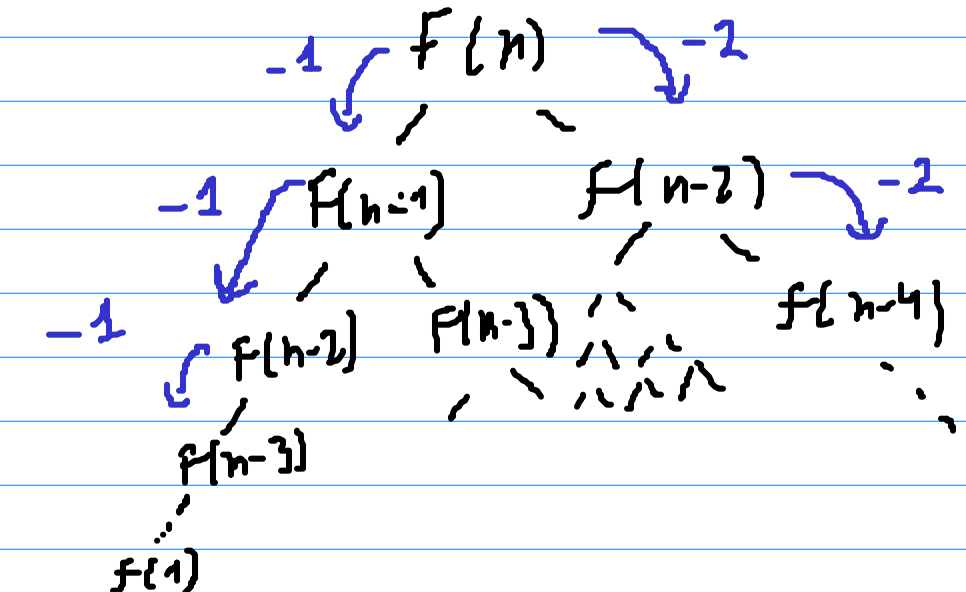
$\{1\}_m$

$\left\{ \frac{1-\sqrt{5}}{2}, \frac{1+\sqrt{5}}{2}, 1 \right\}_m$

$A \cdot \left(\frac{1-\sqrt{5}}{2}\right)^n + B \cdot \left(\frac{1+\sqrt{5}}{2}\right)^n + C \cdot 1^n$

$\Theta\left(\left(\frac{1+\sqrt{5}}{2}\right)^n\right)$

$1,618^n$



сложността е броят върхове в дървото от 0

$$2^{\frac{n}{2}} \leq 1,6^n \leq 2^n$$

Зад 5 $T(n) = T(n-1) + 1$

$1 \in \Theta(1)$

характеристично уравнение

$T(n) - T(n-1) = 1$

$\lambda - 1 = 0$

корени

$\{1\}_m \cup \{1\}_n$

n 0 n

1

$\{1,1\}_m$

$$T(n) = A n^0 1^n + B n^1 1^n$$

$$= A \cdot 1 + B \cdot n = \Theta(n)$$

Зад 6

$$T(n) = 3T(n-1) + 4T(n-2) + \underbrace{1}_{\Theta(1)}$$

$$\underbrace{T(n) - 3T(n-1) - 4T(n-2)}_{\lambda^2 - 3\lambda - 4} = 1$$

$$\searrow \quad n^0 \cdot 1^n$$

$$\lambda_1 = -1 \quad \lambda_2 = 4 \qquad \{1\}_n$$

$$\cup$$

$$\{-1, 4\}_n \qquad \{-1, 1, 4\}_n$$

$$T(n) = A(-1)^n + B\,1^n + C\,4^n$$

$$\Theta(4^n)$$

Задача 3

$$T(n) = T(n-1) + T(n-2) + T(n-3) \dots T(0) =$$

$$= \sum_{i=0}^{n-1} T(i)$$

$$T(n-1) = T(n-2) + T(n-3) \dots T(0) + 1$$

$$T(n) - T(n-1) = T(n-1)$$

$$T(n) - 2T(n-1) = 0$$

$$\lambda - 2 = 0$$

$$\lambda = 2$$

$$T(n) \approx \Pi \cdot 2^n \Rightarrow \quad \Theta(2^n)$$

Задача

Сложността на func(n,n)

- На всяка стъпка имаме ТОЧНО две рекурсивни извиквания.
- На всяко рекурсивно извикване намаляваме ТОЧНО един от двата параметъра
- Едното е когато двата параметъра са 0

$R(0,0) = \Theta(1)$

$R(0,j) = 2R(0,j-1) + 1 \rightarrow T(k) = 2T(k-1) + 1$ ✓

$R(i,0) = 2R(i-1,0) + 1 \rightarrow T(k) = 2T(k-1) + 1$ ✓

$R(i,j) = R(i-1,j) + R(i,j-1) + 1 \rightarrow T(k) = 2T(k-1) + 1$ ✓

Полагаме $k = i + j$

$T(k) = 2T(k-1) + 1$

$T(k) - 2T(k-1) + 1$

$\lambda - 2 = 0$ → $\{1\}_m$

$\lambda = 2$ ∪

$\{2\}_m$ $\{1,2\}_m$

$T(k) = A \cdot 1^k + B \cdot 2^k$ $\Theta(2^k)$.

func(n, n) → $i + j = 2n$ $\Theta(2^{2n}) =$

$= \boxed{\Theta(4^n)}$

3. сд 10 $T(0) = 1$

$$T(n) = T(n-1) + 2^n$$

10.1 решение с разбиване

$T(n-1) = T(n-2) + 2^{n-1}$

$T(n-2) = T(n-3) + 2^{n-2}$

$\vdots$

$T(0) + 2^1$

1

$$2^n + 2^{n-1} + 2^{n-2} \dots + 2^1 + 1$$

$$= 2^{n+1} - 1 \qquad \boxed{\Theta(2^{n+1})} \quad (= \Theta(2^n))$$

10.2 решение с характ. уравнение

$$T(n) = T(n-1) + 2^n$$

↓ ↓ $n^0 (2)^n$

$\lambda - 1 = 0$ $\cup$ $\{2\}_m$

$\{1\}_m$

$\{1, 2\}_m$

$T(n) = A \cdot 1^n + B \cdot 2^n$ $\boxed{\Theta(2^n)}$

! $\Theta(2^{n+1}) = \Theta(2^n)$

$2^{n+1} = \underline{2} \cdot 2^n$

КОНСТАНТА