DAA семинар 7

Зад Да се реши уравнението: $T(n) = 2T(\frac{n}{2}) + \frac{n}{\lg n}$

Решение: $k = \log_b a = \log_2 2 = 1$

Сравн. $n^k = n^1$ с $\frac{n}{\lg n} = f(n)$

I сл. ?$\exists \varepsilon > 0$: $n^{1-\varepsilon} \succeq \frac{n}{\lg n} \iff \frac{n}{n^\varepsilon} \succeq \frac{n}{\lg n} \iff n^\varepsilon \preceq \lg(n)$

За $\forall \varepsilon > 0$ $n^\varepsilon > \lg(n) \Rightarrow \forall \varepsilon > 0$ $n^{1-\varepsilon} \not\succeq \frac{n}{\lg(n)}$

II сл. $\exists ? t \geq 0, t \in \mathbb{Z}$: $n^1 . \lg^t(n) \asymp \frac{n}{\lg n} \iff \exists t \geq 0$: $\lg^{t+1}(n) \asymp 1$

$\Rightarrow$ Не, става при $t+1=0$, т.е. $t=-1$, но $t \geq 0$

III сл. ?$\exists \varepsilon > 0$: $n^{1+\varepsilon} \preceq \frac{n}{\lg n} \iff n^\varepsilon \preceq \frac{1}{\lg n} \iff n^\varepsilon \lg n \preceq 1$

$\Rightarrow$ Не; $\lim\limits_{n \to \infty} \frac{n^\varepsilon \lg n}{1} = +\infty \Rightarrow n^\varepsilon \lg n > 1$ $\forall \varepsilon > 0$

$\Rightarrow n^{1+\varepsilon} \not\preceq \frac{n}{\lg n}$

$\Rightarrow$ Не можем да го решим с МТ.

Сравнението става $\Rightarrow$ DP

Метод на Акra-Bazzi

Нека $T(n) = \sum\limits_{i=1}^{n} (a_i T(b_i n + h_i(n))) + g(n)$, където

за $i = 1..n$: 1) $a_i > 0$ са константи

2) $b_i \in (0, 1)$ са константи

3) $h_i \leq \frac{n}{\lg^2(n)}$

4) $g'(n) \preceq n^c$ за някаква константа c

5) $\exists p : \sum a_i b_i^p = 1$ $\Rightarrow T(n) \asymp n^p \left(1 + \int\limits_1^n \frac{g(u)}{u^{p+1}} du\right)$

Зад $T(n) = \frac{7}{4}T(\frac{n}{2}) + T(\frac{3n}{4}) + n^2$

Решение: $p = ?$, т.е. $\frac{7}{4}\cdot(\frac{1}{2})^p + 1\cdot(\frac{3}{4})^p = 1$

търсим по-голямо

Пробване различни p: $p=1$: $\frac{7}{4}\cdot\frac{1}{2}+\frac{3}{4} > 1$ → не става

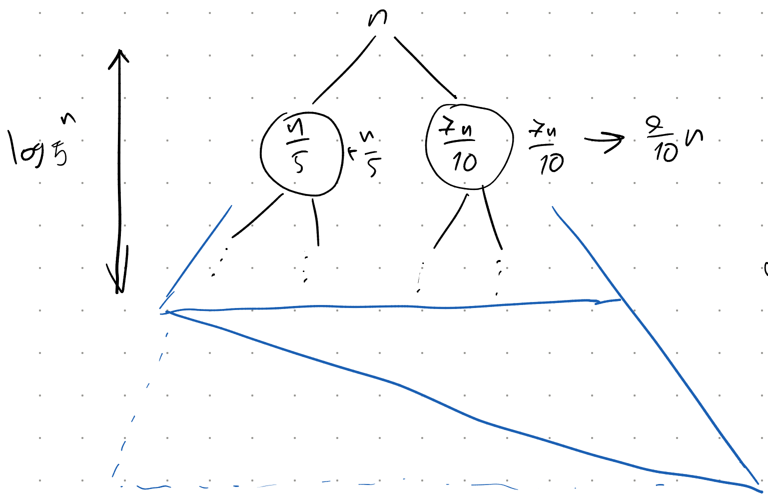
$p=2$: $\frac{7}{4}\cdot\frac{1}{4} + 1\cdot\frac{9}{16} = \frac{7+9}{16} = 1 \Rightarrow p=2$

$$\Rightarrow T(n) \asymp n^2\left(1+\int_1^n \frac{u^2}{u^{2+1}}du\right) = n^2\left(1+\int_1^n \frac{1}{u} du\right) = n^2\left(1+\ln(u)\Big|_{u=1}^{n}\right) =$$

$$= n^2(1+\ln(n)-\ln(1)) = n^2 + n^2\ln(n) \asymp n^2\lg(n)$$

Зад Да се реши $T(n)=T(\frac{n}{5})+T(\frac{7n}{10})+n$, което описва сложността по време на алгоритъма Median of Medians.

Решение: I н. Дърво на рекурсията



Всяко ново ниво има тежест $\frac{9}{10}$ от тежестта на предишното

Дървото не е пълно $\Rightarrow$ тежестта му $T(n)$ е

$$T(n) \le \sum_{i=1}^{\infty} \left(\frac{9}{10}\right)^i n =$$

$$= n\cdot\sum_{i=1}^{\infty}\left(\frac{9}{10}\right)^i = \Theta(n)$$

Но и $T(n) \ge n$ $\Rightarrow$ $T(n) = \Theta(n)$

Това трябва да се докаже по индукция

II н. Акра-Бази:

Търсим p: $\left(\frac{1}{5}\right)^p + \left(\frac{7}{10}\right)^p = 1$

$p=1 \to \frac{1}{5}+\frac{7}{10} < 1$ → търсим по-малко

Нека $p \in (0;1)$

$$T(n) = n^p\left(1+\int_1^n \frac{u}{u^{1+p}}du\right) = n^p\left(1+\int_1^n u^{-p}du\right) \overset{-p\neq -1}{=} n^p\left(1+\frac{u^{1-p}}{1-p}\Big|_{u=1}^{n}\right) =$$

$$= n^p\left(1 + \frac{n^{1-p}}{1-p} - \frac{1^{1-p}}{1-p}\right) = \underset{(0;1)}{n^{p}} + \frac{n}{1-p} - \frac{1^{1-p}}{1-p}\cdot \underset{\in(0,1)}{n^{p}} = \Theta(n)$$

Основни сортировки

Име	T(n)	S(n)	Забележка
Bubble sort	n^2	1	
Selection	n^2	1	най-малко размествания
Insertion	n^2	1	стабилен; най-добър за малки n

// ↑ Тези с т.нар. наивни сортировки. Следващите са бързи

Merge	$n \lg n$	n	може да се паралелизира
Heap	$n \lg n$	1	in-place; нестабилен → може да се направи стабилен, но повече T и S
Quick	$n \lg n$ n^2 в най-лош	$\lg n$ n най-лош	най-добър на практика

// Quick + Heap = IntroSort

Counting sort	$n+k$	k	Ако знаем, че елементите имат най-много k възможни стойности
Radix sort	n - на теория $n \log_b n$	1	Само за числа; b - основа на бройната система
Bogo sort	$n!$		Любимата сортировка по ЛП...

Когато не знаем какво да правим с данните, може да ги сортираме

Двоично търсене

```
1. int binary_search(int arr[], int n, int val) {
2.   int left = 0, right = n - 1;
3.   while (left <= right) {
4.     int mid = left + (right - left) / 2;
5.     if (arr[mid] == val)
6.       return mid;
7.     if (arr[mid] < val)
8.       left = mid + 1;
9.     else
10.      right = mid - 1;
11.  }
12.  return -1;
13. }
```

Ще ни върне индекс, на който се намира val, ако няма такъв −1

<u>Инвариант</u>: При всяко достигане на ред 5 val не се намира в arr[0... left−1] и в arr[right+1 .. n−1]

// Може и: val се съдържа (ако е така) в arr[left ... right]

• База: При първото дост. на ред 5, left = 0 и right = n−1

Наистина val не се намира в arr [0.. 0−1] и в arr [n−1+1 ... n−1]

• Поддръжка: Нека инв. е верен за някое дост. на ред 5, което не е последно

Тогава left ≤ right и val не се намира в arr[0... left−1]
arr[right+1 ... n−1]

$mid = \left\lfloor \frac{left + right}{2} \right\rfloor$ и $left \le mid \le right$

<u>1сл.</u> arr[mid] = val − не може; не е посл. достигане

<u>2сл</u> arr[mid] < val: Понеже arr е сортиран, val $\notin$ arr[0...mid]

Оттук val $\notin$ arr $[0 \ldots \underbrace{mid+1}_{left'} -1]$ и val $\notin$ arr[right+1 ... n−1]

<u>3сл</u> arr[mid] > val − аналогично на 2сл.

• Терминация: Излизаме или при left > right, или при arr[mid] = val

<u>1сл</u> arr[mid] = val → коректно връщаме mid

2сл left > right. Тогава left−1 ≥ right и от инв. имаме, че

val ∉ arr [0... left−1] и val ∉ arr[right+1...n−1]

⇒ val ∉ arr [0..n−1] и връщаме −1, което трябва

Зад Даден е масив A[1..n], съдържащ числата от 0 до n в сортиран вид, като едно от тях липсва. Кое е то?

Решение: Iн. Линейно обхождане → $T(n)=\Theta(n)$

IIн. Двоично търсене: Спрямаме за съответствието м/у число и индекс − до едно място съответстват, след това са различни.

$T(n) = \Theta(\lg n)$

Допълнение към задачата: Ако A не е сортиран?

Iн. Сортиране + двоично търсене → $T(n)=O(n\lg n)+O(\lg n) = O(n\lg n)$

IIн. Броене кое число се среща → $T(n)=\Theta(n)$ $S(n)=\Theta(n)$

// Ако може да модифицираме A: За ∀срещнато число k:

$A[|k|] = -A[|k|]$

Така търсеното число е индекса на първото положително.

IIIн. $\sum_{i=0}^{n} i - \sum_{i=1}^{n} A[i] = \frac{n(n+1)}{2} - \sum_{i=1}^{n} A[i] \rightarrow T(n)=\Theta(n)$

Зад. Произвеждане крушки. Имаме висока сграда, от която да ги хвърляме. Искаме да разберем от каква височина се чупят.

− Макс. височина без счупване, най-малко хвърляния

⇒ Двоично търсене

− Макс. височина, най-малко счупвания:

→ Линейно нагоре от 1-вия етаж

Много пускания, но ≤1 счупване

- Имаме 2 крушки, като счупим не може да ползваме повече
  Искаме min брой хвърляния

I н. Двоично търсене с първата + линейно с втората

В най-лошия случай $\frac{n}{2}$ хвърляния

II н Линейно търсене със стъпка $\sqrt{n}$

$\frac{n}{\sqrt{n}} = \sqrt{n}$ най-много хвърляния с първата

+ наивно търсене в интервалът с дължина $\sqrt{n}$, който сме намерили.

$2\sqrt{n} \leq \frac{n}{k}$ $k=2$ за двоично търсене

// TieredVector