ДАА

упр.№9, 14 май 2020г.

Hello, world!

Зад.4: **(3SUM)** В масив A[1..n] без повтарящи се числа да се намерят три числа със сума 0

- I вариант: 3 вложени цикъла $\rightarrow T(n) = \Theta(n^3)$; $M(n) = \Theta(1)$
- II вариант: сортиране + за всяка двойка индекси i, j търсим двоично дали $-(A[i]+A[j])$ присъства в масива

$\rightarrow T(n) = \underbrace{\Theta(n \lg n)}_{\text{сорт.}} + \Theta(n^2) \cdot O(\lg n) = O(n^2 \lg n)$

$M(n) = \Theta(1) + \Theta(1) = \Theta(1)$

- III вариант - ~~няма~~ че сортираме, аналогично търсим
- ($A[i] + A[j]$) сам е в масива, но това е чрез хеш
(първо вмъкване всички ел-ти в хеш)

$\Rightarrow$ очакваното време за търсене е $\Theta(1) \Rightarrow$ очаквано общо $\Theta(n^2)$

$\Rightarrow$ най-лош случай $\Theta(n^3)$
(също, $M(n) = \Theta(n)$)

O(n^2) алгоритъм за **(3SUM)**:

```
Sort(A[1..n])
for i ← 1 to n-2
  if A[i] ≥ 0
    return
  j ← i + 1; k ← n
  while j < k
    if A[i]+A[j]+A[k] > 0
      k ← k - 1
    else if A[i]+A[j]+A[k] < 0
      j ← j + 1
    else
      print i,j,k
      j ← j + 1, k ← k - 1
```

$$T(n) \le \sum_{i=1}^{n-2}(n-i) = \Theta(n^2)$$

$$\Rightarrow T(n) = O(n^2)$$

$$T(n) \ge \sum_{i=1}^{n-2}\left(\frac{n-i}{2}\right) = \Theta(n^2)$$

$$\Rightarrow T(n) = \Omega(n^2)$$

$$\Rightarrow T(n) = \Theta(n^2)$$

(3SUM) варианти:

- да се намерят три числа със сума дадено C → ще намалим всички числа с $\frac{C}{3}$ и ще използваме (3SUM)
- да се намерят три числа така, че сумата на две от тях да е равна на третото → ???
- да се намери по едно число от три отделни масива със сума 0
  → нека $A[1..n]$, $B[1..m]$, $C[1..p]$ → конкатенираме масивите и заместваме числата
  $\times 10 + 1$ $\quad \times 10 + 2$ $\quad \times 10 - 3$
- **(4SUM)**
  ↳ n^3 ; $n^2 \lg n$

Зад.5: В масив A[1..n] има предоминиращ елемент (среща се на >50% от позициите). Кой е той?

- сортиране + търсене на най-дълга последователност от еднакви $\Rightarrow T(n) = \Theta(n\lg n) + \Theta(n) = \Theta(n\lg n)$
- сортиране + взимане ел-та по средата $\Rightarrow T(n) = \Theta(n\lg n)$
- $\mathrm{PICK}\left(\frac{n}{2}\right) \Rightarrow$ това връща средния ел-т $\Rightarrow T(n) = \Theta(n)$
- Със стек, в който вкарваме само еднакви ел-ти (иначе pop) $\rightarrow$ вж. псевдокод
  $\Rightarrow T(n) = \Theta(n)$; $M(n) = \Theta(n)$
- Заместване стека с брояч, отговаряне на големината + една променлива за върха му
  $\Rightarrow M(n) = \Theta(1)$

```
1. S ← празен
2. push(S, A[1])
3. for i ← 2 to n
     if A[i] = top(S)
       push(A[i], S)
     else pop(S)
       if empty(S) push(A[i], S)
return top(S)
```

Зад.6: Да се намери сечението/обединението/разликата на две множества, представени като сортирани масиви A[1..n] и B[1..m]

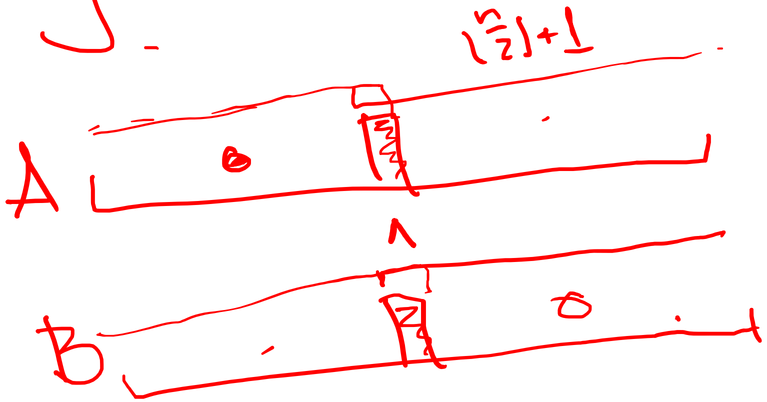
За сечение
- за всеки ел-т от единия масив търсим двоично дали се среща в другия. Нека $n<m$: $T(n,m) = n.\lg m$
- аналог. на Merge: два "итератора"; започват от A[1] и B[1]
  - ако сочат към разл.; преместване итератора, сочещ към по-малкия
  - иначе вмъкване тази ел-т в резултата и инкрементиране и двата
  - като някой ит. достигне края, приключване

За обединение:
- ако сочат към разл.; вмъкване по-малкия и преместване ит.
- иначе вмъкване тази ел-т веднъж и инкр. и двата
- като някой ит. достигне края, вмъкване всички оставащи в резултата

Зад.7: Да се намери общата медиана на два сортирани масива с еднакъв размер A[1..n] и B[1..n]

– заделяме памет за 2n ел-та, вътре Merge(A,B) ⇒ $T(n)=\Theta(n)$

$M(n)=\Theta(n)$

– 

Нека медианата на A е по-малка от тази на B.

⇒ Никой ел-т от $A[1..\lceil \frac{n}{2} \rceil]$ няма да бъде търсения ел-т.

(ще има поне n+1 ел-та по-големи от него)

Аналог. отхвърляме по-голямата половина от B

$T(n)=T(\frac{n}{2})+1$

⇒ $k=\log_b a=0$ ⇒ сравн. $n^0=1$ с $f(n)=1$

⇒ по II сл на МТ $T(n)\asymp \lg n$

⇒ можем да се извикаме рекурсивно за останалите половини на A и B

Зад.8: Всички без едно число в масив се повтарят по $\overset{k}{\text{два}}$ пъти – последното число се среща само $\overset{p}{\text{веднъж}}$. Кое е то?

hint¹: броене (не на самите числа)

hint²: XOR

→ Отказване n / nlgn решения