HW

a) $T(n) = T\left(\frac{9.n}{10}\right) + n$

b) $T(n) = 16.T\left(\frac{n}{2}\right) + n^2$

c) $T(n) = 8.T\left(\frac{n}{2}\right) + n^3$

d) $T(n) = T(n-1) + \sqrt{n}$

e) $T(n) = 2.T(\sqrt[4]{n}) + \log(\sqrt[4]{n})$

f)
```
sum(A[1..n]: array of nums): num
if n=0 then
    return 0
else if n = 1 then
    return A[1]
return sum(A[1..⌊n/2⌋]) +
          sum(A[⌊n/2⌋+1...n])
```

Формулирайте рекурентното уравнение и го решете.

---

a) $a = 1$, $b = \frac{10}{9}$, $k = \log_b a = 0$

$n^k = n^0 = 1$

За $\varepsilon = 0{,}1$, то $n^{k+\varepsilon} \prec f(n) = n$

Тогава по сл. III $T(n) \asymp f(n) = n$

Дали е регулярна n?

$(\exists c \in (0;1))(\exists n_0 \in \mathbb{N})(\forall n \in \mathbb{N})$

$[n \geq n_0 \Rightarrow a \cdot f(\frac{n}{b}) \leq c \cdot f(n)]$

$$\frac{9 \cdot n}{10} \leq c \cdot n \quad // : n \neq 0$$

$$(0;1) \ni \frac{9}{10} \leq c$$

За $c = \frac{9}{10}$ и $\underline{n_0 = 1}$

b) $T(n) = 16 \cdot T\left(\frac{n}{2}\right) + n^2$

$a = 16$, $b = 2$, $k = \log_b a = 4$

$n^k = n^{4-\varepsilon} \succ n^2 = f(n)$

$\varepsilon = 0,1$ I-ви сл.

Така $T(n) \asymp n^k$

c) $T(n) = 8 \cdot T\left(\frac{n}{2}\right) + n^3$

$a = 8$, $b = 2$, $k = \log_b a = 3$

$n^k = n^3 \asymp f(n) = n^3$

По II-ри сл. $T(n) \asymp n^k \cdot \log n = n^3 \cdot \log n$

d) $T(n) = T(n-1) + \sqrt{n}$

Чрез развиване:

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

$T(n) = T(n-2) + \sqrt{(n-1)} + \sqrt{n} = \ldots =$

$= T(0) + \sum_{i=1}^{n} \sqrt{i} = T(0) + \sum_{i=1}^{n} i^{1/2} \simeq n^{1/2+1} =$ $\quad 1/2 > -1$

$= n^{3/2}$

e) $T(n) = 2 . T(\sqrt[4]{n}) + \log(\sqrt[4]{n})$

$n^{1/4} = 4^{\frac{\log n}{4}}$

$T(4^{\log n}) = 2 . T(4^{\frac{\log n}{4}}) + \log(4^{\frac{\log n}{4}}) =$

$= 2 . T(4^{\frac{\log n}{4}}) + \frac{\log n}{4} . \log 4$

Нека $m = \log_4 n \to n = 4^m$.

$Q(m) = T(4^m)$

$Q(m) = 2Q\left(\frac{m}{4}\right) + \frac{m}{4} . \log 4$

$a = 2, b = 4$

$k = \log_4 2 = \frac{1}{2}$, $\underline{m^k = m^{1/2}}$

$f(m) = m . \frac{\log 4}{4}$

За $\varepsilon = 0,1$, то $m^{k+\varepsilon} \precsim f(m)$

Регулярна ли е?

$(\exists c \in (0;1))(\exists n_0 \in \mathbb{N})(\forall n \in \mathbb{N})$
$[n \geq n_0 \Rightarrow a \cdot f(\frac{n}{b}) \leq c \cdot f(n)]$

$$\frac{2 \cdot n \cdot \log^4 n}{4 \cdot 4 \cdot 2} \leq c \cdot \frac{n \cdot \log^4 n}{4} \quad : n \log^4 n$$

$$(0;1) \ni \frac{1}{2} \leq c$$

за $c = 1/2$ ✓, $n_0 = 1$.

по III сл. $Q(n) = f(n) \asymp n$

Така $T(n) \asymp \log n$

зарради sum(..)+sum(..)

f) $T(n) = 2T(\frac{n}{2}) + 1$, $k = \log_2 2 = 1$

$f(n) = n^0 = 1 \prec n^1 = n^k$ за $\varepsilon = 0{,}1$ например става

$1^{ви}$ случай на МТ

Значи $\Theta(n) \asymp n^k = n$

Unfinished work for next time

a) $T(n) = 3 \cdot T\left(\frac{n}{2}\right) + n \cdot \log n$

b) $T(n) = 2 \cdot T\left(\frac{n}{4}\right) + \sqrt{n}$

c) $T(n) = 4 \cdot T\left(\frac{n}{\sqrt{2}}\right) + n^4$

$\frac{HW}{MT}$

d) $T(n) = 2 \cdot T(n-1) + T(\log_2 n) + n$

Използвайки индукция ще го решим, че $T(n) \approx 2^n$.

e) $T(n) = 3 \cdot T(\sqrt[3]{n}) + \log_3 n$

f) $T(n) = T(\log_2 n) + 1$

a) $T(n) = 3 \cdot T\left(\frac{n}{2}\right) + n \cdot \log n$

$f(n) = n \cdot \log n \prec n^{\log_2 3 - \varepsilon}$, защото $\underline{\log_2 3 > 1}$

Колко трябва да е ε?

$$\lim_{n \to \infty} \frac{n^{\log_2 3}}{n \cdot \log n} = \lim_{n \to \infty} \frac{n^{\log_2 3 - 1}}{\log n} = +\infty$$

зебиши се полиномиално

за $\varepsilon = \frac{\log_2 3 - 1}{2} > 0$ е ок.

$T(n) \asymp n^{\log_2 3}$

---

b) $T(n) = 2 \cdot T\left(\frac{n}{4}\right) + \sqrt{n}$

$n^k = n^{\log_4 2} = n^{1/2}$

$f(n) = n^{1/2} \asymp n^k = n^{1/2}$

$\to$ II-ри случай от МТ $T(n) \asymp \sqrt{n} \cdot \log n$

---

c) $T(n) = 4 \cdot T\left(\frac{n}{\sqrt{2}}\right) + n^4$

$f(n) = n^4 \asymp n^{\log_{\sqrt{2}} 4} = n^{2 \cdot \log_2 4} = n^4$

II-ри случай от МТ $T(n) \asymp n^4 \cdot \log n$

d) $T(n) = 2.T(n-1) + T(\log_2 n) + n$

Чрез индукция ще докажем, че асимптотиката му е това на отгатнатата от нас ф-я.

Ако $T(\log_2 n)$ го няма, директно $T(n) \asymp 2^n$. Съобразяваме, че даже и да го има това събираемо, то има пренебрежително малък принос към сумата, защото логаритъма нараства много по-бавно от изходящата.

Сега вече имаме цел и тя е да покажем, че $T(n) \asymp 2^n$.

Разбиваме я на две подцели:

1) $T(n) \succeq 2^n$ // не пада под 2^n

2) $T(n) \preceq 2^n$ // не надскача 2^n

1) Припомняме си добре зададената деф. $f(n) \succeq g(n) \iff$
$(\exists c>0)(\exists n_0 \in \mathbb{N})(\forall n \in \mathbb{N})$
$[n \geq n_0 \implies 0 \leq c.g(n) \leq f(n)]$

В нашият случай изглежда така:
$(\exists c>0)(\exists n_0 \in \mathbb{N})(\forall n \in \mathbb{N})$
$[n \geq n_0 \implies 0 \leq c.2^n \leq T(n)]$

Ще използваме пълна (силна) математическа индукция.
Разбрахме се, че за base случаи в този курс не се интересуваме често и само ще го пренебрегнем.
(ИП) Допускаме, че има c, т.е.:

$$T(n-1) \geq c.2^{n-1}$$
$$T(n-2) \geq c.2^{n-1}$$
$$\dots$$
$$T(\log_2 n) \geq c.2^{\log_2 n} = c.n$$
$$\dots$$
$$T(1) \geq c.2^1 = c.2$$

Ще използваме ИП само за $T(n-1)$ и $T(\log_2 n)$ + definition of $T(n)$

$$T(n) = 2.T(n-1) + T(\log_2 n) + n \overset{(\text{ИП})}{\geq}$$
$$2.c.2^{n-1} + c.n + n =$$
$$= c.2^n + \underbrace{n.(c+1)}_{\geq 0} \geq c.2^n$$

Т.е. $T(n) \geq c.2^n$ за достатъчно големи n (сега не се интересуваме от конкретното) и сме готови.

2) Сега искаме $T(n) \leq 2^n$. ($T(n) \in O(2^n)$)
От def. търсим $c > 0$, т.е. за достатъчно големи n, то
$T(n) \leq c \cdot 2^n$.
Отново ползваме ПМИ (ползваме и допускаме c, т.е. $c > 0$ и
$T(n-1) \leq c \cdot 2^{n-1}$
$T(n-2) \leq c \cdot 2^{n-2}$

...

$T(\log_2 n) \leq c \cdot 2^{\log_2 n} = c \cdot n$

...

$T(1) \leq c \cdot 2^1 = c \cdot 2$
Отново ползваме само $T(n-1)$ и $T(\log_2 n)$ (ИП) и def. $T(n)$:
$T(n) \leq c \cdot 2^n + c \cdot n + n$
Много ни се иска да е по-малко от дясното страна от $c \cdot 2^n$, но не става т.к. $c \cdot n + n > 0$ за $n \geq 1$.

? $T(n) \le c\cdot 2^n - d\cdot\alpha^n$?

Втори опит? Как да подходим?
Нека се опитаме да заявим (ИП) и то ще е, че за $c, d > 0$ и $\alpha \in (1;2)$, то за дост. големи n имаме:

после свиване заедно

$$T(n-1) \le c\cdot 2^{n-1} - d\cdot\alpha^{n-1}$$
$$T(\log_2 n) \le c\cdot 2^{\log_2 n} - d\cdot\alpha^{\log_2 n} =$$
$$= c\cdot n - d\cdot\alpha^{\log_2 n}$$

Тогава
$$T(n) \le c\cdot 2^n - d\cdot 2\cdot\alpha^{n-1} + c\cdot n - d\,\alpha^{\log_2 n} + n$$

Искаме животнята отдясно да $\le$ от $c\,2^n - d\cdot\alpha^n$, което на свой ред е $\le c\cdot 2^n$.

$$\cancel{c\cdot 2^n} - d\cdot 2\cdot\alpha^{n-1} + c\cdot n - d\cdot\alpha^{\log_2 n} + n \le \cancel{c\,2^n} - d\,\alpha^n$$
$$\Leftrightarrow -2\cdot d\cdot\alpha^{n-1} + c\cdot n - d\cdot\alpha^{\log_2 n} + n \le -d\cdot\alpha^n \Leftrightarrow$$
$$\underbrace{d\cdot\alpha^n - 2\cdot d\cdot\alpha^{n-1}}_{\text{експоненциална}} + \underbrace{c\cdot n + n - d\cdot\alpha^{\log_2 n}}_{\text{линейна}} \le 0$$

има смисъл за $\alpha > 1$

Т.к. абсолютни ст-ти експоненц. доминира линейната, то ще изследваме само експ. Ако тя е отриц. за дост. големи n, следователно

$d\cdot\alpha^n - 2\cdot d\cdot\alpha^{n-1} < 0 \quad // : \alpha^{n-1}\cdot d$

$\alpha - 2 < 0$

$\underline{\alpha < 2}$, което е в сила от (2n), това че е истина.

Значи и това което се опитваме да докажем, че $T(n) \le c.2^n - d.\alpha^n$ е в сила. Така:

$T(n) \le c.2^n - d\alpha^n \le c.2^n$

за $c, d > 0$, $\alpha \in (1;2)$

т.е. $T(n) \preceq 2^n$ ■

е) $T(n) = 3.T(\sqrt[3]{n}) + \log_3 n$

Нека $n = 3^{\log_3 n} \to n^{1/3} = 3^{\frac{\log_3 n}{3}}$

$Q(m) = T(3^m) = 3.T(3^{\frac{m}{3}}) + m$

$m = \log_3 n$

$Q(m) = 3.Q\left(\frac{m}{3}\right) + m$

$k = \log_3 3 = 1 \to m^k = m^1 \asymp f(m) = m$

II сл. МТ $\to Q(m) \asymp m^k.\log_3 m =$

$= m.\log_3 m$

$T(n) \asymp \log_3 n . \log_3 \log_3 n$

f) $T(n) = T(\log n) + 1$

Не ни ли наговиха нещо от първите часове по ДАА.

Колко пъти трябва да приложим log в/у n, т.е. да стане ≤ 1 (?)

Именно $\log^* n$

$n \to \log n \to \log\log n \to \ldots \to \log^* n$

$\Theta(1) \quad \Theta(1) \quad \Theta(1) \quad \Theta(1)$

$\log^* n =$ сбор от $\Theta(1)$

$\log^* n \leq 1$

Така $T(n) \asymp \log^* n$.

Дизайн на алгоритми.
(с масиви)

Зад.1) Разбиване (partition)

Дадено: 1) числов масив A[1...n]
2) критична стойност value

Търси се: разбиване на A на 2 части – големи и малки ел-ти.

За A = [5, 9, 8, 16, 24, 3]; value = 10

small = [5, 8, 3]

big = [16, 24, 3] $T(n) \approx n$.

Зад.2) Търсене на дадена стойност

Дадено: 1) масив A[1..n]
2) стойност key

Търси се индексът i на key т.ч. A[i]=key

Алгоритъм:

- ако масивът е сортиран -> binary search: $T(n) \approx \lg n$
- ако масивът не е сортиран -> последователно търсене: $T(n) \approx n$

Зад 3) Търсене на брой сравнения за:

a) min/max в произволен масив;

b) min и max в произволен масив;

c) Може ли b) да се реализира с по-малък брой сравнения?

d) А да намерим кой a) и втория по големина max/min?

---

a) С едно обхождане на масива $\to n-1$ сравнения $\to T(n) \asymp n$

b) С две обхождания $\to 2.(n-1) \to T(n)=n$.

c) Може да минем и с $\frac{3}{2}.n$ сравнения.

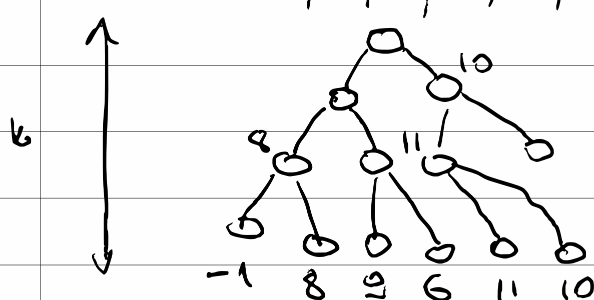
Нека разделим масива мислено на последователни двойки т.е:

$A = [(a_1, a_2), (a_3, a_4), \ldots, (a_{n-1}, a_n)]$

Ако n е нечетно, последния е сам. На тези двойки им взимаме max и min т.е. $n/2$ сравнения за $n/2$ брой двойки.
Най-накрая взимаме max на м-вото от маковете и min на м-вото от min-овете по $n/2$ и $n/2$ сравнения.

Като ги събереш $n/2 + n/2 + n/2 = \frac{3}{2} \cdot n$.

д) Нека съпоставим кореново дърво на „турнира" (борбата за визи кой е I-вия най-голям). По листата ще наслагаме нашите претенденти и т.н. сравненията са м/у двата, то е очевидно (избираме дървото да е с височина и брой листа спрямо брой на елементи в масива -> да е най-малкото дърво, така са отговарящо за нашите претендентите по листата) Т.е. примерно за $A = [-1, 8, 9, 6, 11, 10]$



За I-вия е ясно, че $(n-1)$ сравнения се правят. Нека h е височината на дървото. II-рия може да бъде бит само от I-вия значи го търсим по пътя

на $I^{вия}$ е дължина $k = \lfloor \log_2 n \rfloor$
т.е. ни трябват $\lfloor \log_2 n \rfloor - 1$
сравнения за да го намерим като
знаем $I^{вия}$.
Така за втори обход ни трябва
$(n-1) + (\lfloor \log_2 n \rfloor - 1)$ сравнения.

Зад. 4 <u>Търсене на $k^{ти}$ най-малък / най-голям елемент</u> (вкл. и търсене на медиана)
Алгоритъм PICK с $T(n) \asymp n$
Не е изпитен от къси понят.

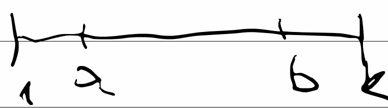
Зад. 5 Имаме масив и искаме да намерим двата елемента, за които $A[i] . A[j]$ е максимално.

Brute force: $T(n) \asymp n^2$
Може ли по-добре?
Не става със задачата за намиране на $I^{ви}$ и $II^{ри}$ max, защото за
$A = [-12, -11, \underbrace{0, 1}_{max-ове}]$, но $-12 . -11 > 0 . 1$

Тогава ще намерим 2та най-големи и 2та на най-малки елемента и ще направим произведението им и накрая от двете произведения взимаме max-ото им.

$((n-1) + (\lfloor \lg n \rfloor - 1)) . 2 + 1$

Зад 6 Имаме масив $A[1...n]$ от естествени числа в интервала: $1 \le a_i \le k$ за $i \in \{1, ..., n\}$.
Дадено е и това k.
Колко числа има в интервала $[a, b]$ за $1 \le a \le b \le k$, $a, b \in \mathbb{N}$?



Зад 7 Даден е сортиран масив с числата от 0 до n и едно от тях липсва. Кое е то?

Зад 8 Имаме два сортирани масива $A[1...n]$ и $B[1...n]$ с еднакъв брой елементи. Имаме $C = A \cup B$.

union

$\boxed{HW}$ Имаме $2n$ и
масив, в който на
е силата на играч i.
Да се разделят на два отбора по n
играча, т.е. да са максимално
неравностойни.
Hint: Оптимален времеви алгоритъм е
със сложност $\Theta(n)$.