Задача 1

Даден е сортиран масив arr = [a1, a2 ..., an]. a1 <= a2 <= a3 ... <= an.

Алгоритъм, който връща дали всички елементи са уникални.

Compare(x, y) -> <, >, =.

Докажете "добра" долна граница за точния брой сравнения, които алг ще извърши.

Решение: Ще покажем, че **n - 1** сравнения е долна граница.

Допускаме, че съществува алгоритъм Alg, който с **n - 2** сравнения връща верен резултат.

Да разгледаме **X = [x1, x2 ... xn]** xi = 2 * i

Alg(X) -> **истина** (всички елементи са уникални).

Знаем, че Alg извършва **n - 2 сравнения**.

-> Съществува поне един елемент xi, който не е сравнен със следващия (x_i+1).

Да разгледаме **X'** = [x1, x2 ...xi, xi ... xn] *(xi+1 = xi).*

X' е сортиран!

Alg(X') -> истина (Alg не е сравнил xi с xi+1)

Но това не е вярно! **В X' има повтарящи се елементи**!

-> Alg не е коректен алгоритъм.

-> Не съществува КОРЕКТЕН алгоритъм (с директни сравнения), който извършва < n-1 сравнения.

Задача 2

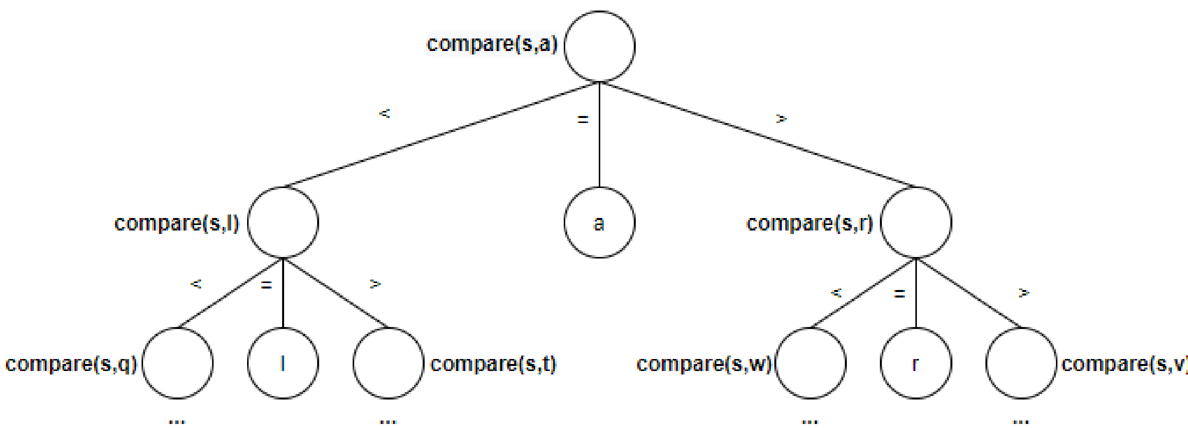
Даден е сортиран масив arr = [a1, a2 ... an]. a1 <= a2 ... <= an.

Намерете и докажете долна граница на брой сравнения за алгоритъм, който търси елемент в масива.

Използваме само сравнения от видя: Compare(x, y) връща **<** или **>** или **=** .

Alg(searched, array)

Да разгледаме дървото на вземане на решения (**decision tree**):



Наблюдение 1: За всяко едно сравнение (извикване на compare) съответства едно листо (което е елемент на масива). Понеже имаме n елемента на масива, то трябва да имаме по n такива листа, от където следва, че трябва да имаме поне n сравнения (извиквания на compare).
Наблюдение 2: На i-тото ниво имаме най-много 2^i сравнения.

Наблюдение 3: На всяко сравнение отиваме едно ниво надолу в дървото.

Следователно търсим височината на дървото (k) !
Броят сравнения по нивата (от наблюдение 2):

$$2^0 + 2^1 + 2^2 + \ldots + 2^{k-1} = \sum_{i=0}^{k-1} 2^i$$

Което знаем, че е 2^k – 1. От **наблюдение 1**:

$$\begin{aligned} 2^k - 1 &\geq n \\ 2^k &\geq n + 1 \\ \log 2^k &\geq \log(n+1) \\ k &\geq \log(n+1) \end{aligned}$$

Следователно имаме долна граница на сравненията **log(n + 1)**

Задача 3

Даден е сортиран масив arr [a1 ... an]. Търсим елемент **searched** в масива. Позволени са само въпроси от вида:

CompareAdvanced(x,y,z,s) (x < y< z)=

1. s < x

2. s = x

3. x < s < y

4. s = y

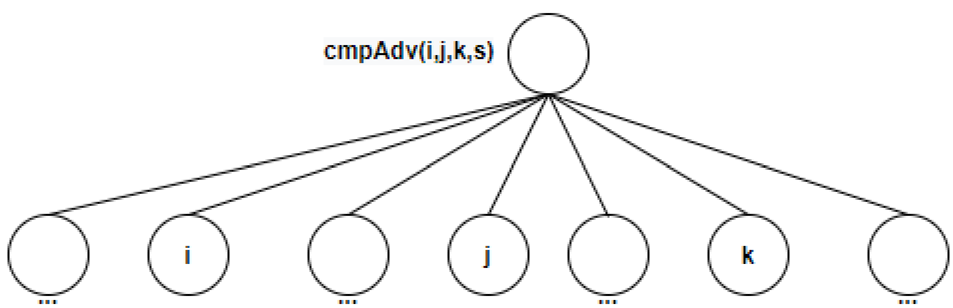
5. y < s < z

6 s = z

7. z < s

Докажете долна граница за брой сравнения от този тип: [log(n+1)]/2

Да разгледаме дървото на вземане на решения (**decision tree**):



Наблюдение 1: За всяко едно сравнение (извикване на **compareAdvenced**) съответстват **ТРИ листа** (които е елемент на масива). Понеже имаме n елемента на масива, то трябва да имаме по n такива листа, от където следва, че трябва да имаме поне n/3 сравнения (извиквания на **compareAdvenced**).
Наблюдение 2: На i-тото ниво имаме най-много 4^i сравнения.

Наблюдение 3: На всяко сравнение отиваме едно ниво надолу в дървото.

Следователно броят на сравненията в дървото са:

$$4^0 + 4^1 + 4^2 + \ldots + 4^{k-1} = \sum_{i=0}^{k-1} 4^i,$$

Което е (4^k -1)/3. От **наблюдение 1** имаме:

$$\begin{aligned}\frac{4^k-1}{3} &\geq \frac{n}{3}\\ 4^k-1 &\geq n\\ 4^k &\geq n+1\\ \log_4 4^k &\geq \log_4(n+1)\\ k &\geq \log_4(n+1)\\ k &\geq \log(n+1)/2\end{aligned}$$

Задача 4

X = [x1, ... xn] и Y = [y1 ... yn]. X и Y са **сортирани**.

Дайте долна граница на броя сравнения за сливане на двата масива (**merge**).

Решение:

Ще докажем следната долна граница**: 2n-1 сравнения.**

Да допуснем, че съществува алгоритъм **Alg**, който слива двата масива с **2n-2** сравнения.

Alg(X,Y) = Z – използвайки 2n - 2 сравнения.

Нека **X = [1, 3, 5 ...] xi = 2*i - 1** и нека **Y = [2, 4, 6 ...] yi = 2*i**

Alg(X,Y) = [1,2,3,4,5,6....] = [x1 , y1, x2, y2, ,x3, y3 ....].

-> съществува елемент xi, който не е сравнен с yi или с yi-1.

1сл. Нека xi не е сравнен с yi. Да разгледаме:

X' = [1 ,3 , 5 ... xi -> xi+1]   Y' = [2 ,4, 6 ... yi -> yi-1 ...]

X' и Y' са сортирани!

Alg(X', Y') = [x1, y1, x2, y2, x3 , y3 ...] xi преди yi -> **масивът не е сортиран!**

-> **алгоритъмът НЕ връща верен резултат!**

2сл. Аналогично!

=> Не съществува **КОРЕКТЕН** алгоритъм (директни сравнения), който да слее два списъка за по-малко от **2n-1** сравнения

Задача 5

Дадени са n телефона и n калъфа. На всеки телефон пасва точно един калъф!

Не са позолени сравнения между два телефона и два калъфа!

Позволени са само сравнения от вида:

compare(телефонX, калъфY) =

1. калъфY пасва точно на телефонX

2. калъфY е голям за телефонX

3. калъфY е малък за телефонX.

Докажете, че всеки алгоритъм, който намира кой телефон на кой калъф съотвества работи във време \Omega(n.log(n))

Решение:

Пасването на телефоните с калъфите е биекция. От множеството на всички биекции

от множеството на телефоните към множеството на калъфите,

трябва да намерим конкретната биекция.

Биекциите са **n!** - възмоните изходи. Следователно дървото на взема на решения има поне n! листа. Разклонеността е 3. Следователно височината на дървото е поне log_3 (n!), което е

\Omega (nlog(n)).

Задача 6
Всеки сортиращ алгоритъм, базиран на директни сравнения, имат долна граница на работа \Omega (n*log(n))

Решение:

Сортирането е търсенето на тази предмутация на елементите, в която те са сортирани.
Тези пермутации са n!. Тук разсъжденията са същите като в горната задача!.