

e) elementarnimi upravami do vzorce dle definice se dostaneme na

$$2^{2^n} = O(2^n)$$

$$2^{2^n} \leq c \cdot 2^n$$

$$(2^n)^2 \leq c \cdot 2^n$$

$$2^n \leq c$$

což neplatí, neboť konstanta nemůže být větší než exponenciála. Tvrzení tedy neplatí.

$$2^{2^n} \neq O(2^n)$$

g)

$$n! = O(n!)$$

$$2^{\log(n+1)} = \frac{1}{5} \cdot 10^{\log(n+1)} = \frac{1}{5} \cdot (n+1) = O(n+1)$$

$$\sqrt{\log n} = O(\sqrt{\log n})$$

$$\log \sqrt{n} = \log n^{\frac{1}{2}} = \frac{1}{2} \cdot \log n = O(\log n)$$

$$n^n = O(n^n)$$

pak

$$\sqrt{\log n} < \log \sqrt{n} < 2^{\log(n+1)} < n! < n^n$$

h)

$$2^{2^n} = O(2^{2^n})$$

$$2^{\log_8 n} = n^{\log_8 2} = n^{\frac{1}{3}} = O(\sqrt[3]{n})$$

$$\sqrt{n} = O(\sqrt{n})$$

$$2^{2^n} = O(4^n)$$

$$n \cdot 2^n = O(n \cdot 2^n)$$

pak

$$2^{\log_8 n} < \sqrt{n} < 2^{2^n} < n \cdot 2^n < 2^{2^n}$$

4)

Tahle uloha je moc pekna a opravdu ji lze resit v lineární prostorové složitosti, tedy $O(1)$. Pokusím se popsat řešení.

Jestliže víme, že jeden prvek se vyskytuje více než n -půlkrát, pak si můžeme analogicky představit ulohu jako toto:

Na začátku máme dva prvky, A a B . Porovnáme je mezi sebou, a jestliže $A \neq B$, pak je můžeme oba “zahodit” a pro A a B zvolit dva následující prvky z rady. Proč? Protože jestliže porovnáme mezi sebou takto $n/2$ prvků a pro všechny porovnání bude platit $A \neq B$, pak na zbydou dva prvky, které se rovnají. Tedy $A = B$. A to je náš hledaný prvek.

Můžeme si to představit zjednodušeně pro konečný počet prvků, třeba $n = 100$. Jestliže $n/2 = 50$, tak prvek se nám bude v řadě opakovat nejméně 51krát. Proto my vykonáme v nejhorším případě 49 porovnání, kde bude A rozdílné od B a v konečném, 50. porovnání, bude $A = B$.

V Python kodu si to můžeme představit takto:

```
def findMajority(arr, n):

    maxCount = 0
    index = -1
    for i in range(n):

        count = 0
        for j in range(n):

            if(arr[i] == arr[j]):
                count += 1

        # updatuj maxCount, jestliže je
        # stavajici element větší
        if(count > maxCount):

            maxCount = count
            index = i

    # jestliže je maxCount > n/2, vrat odpovídající prvek
    if (maxCount > n//2):
        print(arr[index])

    else:
        print("No Majority Element")
```