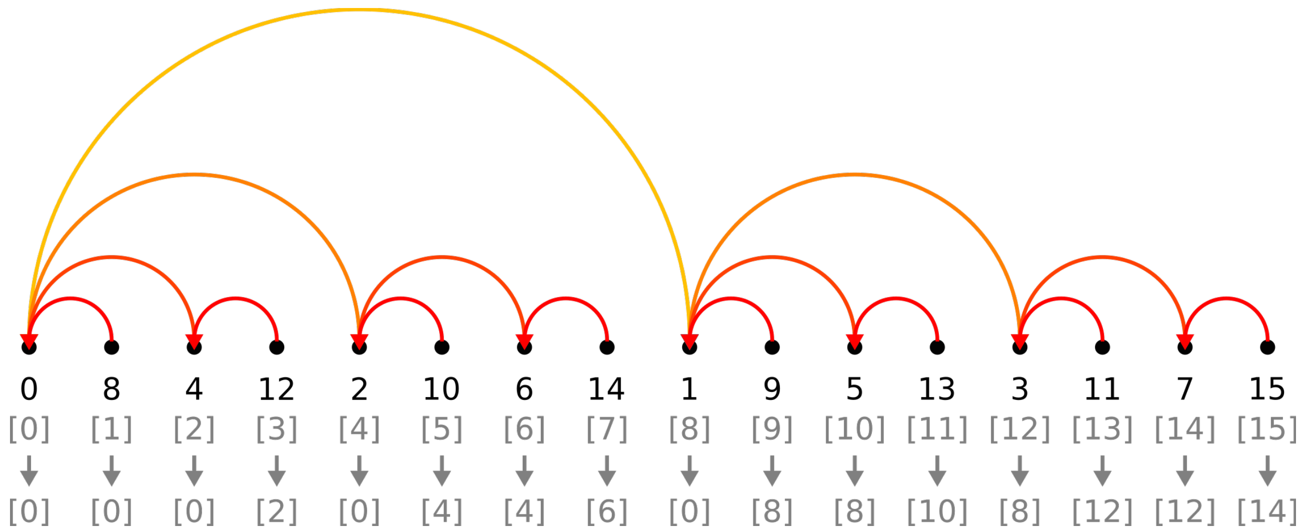
SP1 – 1. teoretická sada

Jméno:	Pavel ŠIGUT
Osobní číslo:	A16B0141P
Počet hodin:	cca 10

1.

a)



A = {0, 8, 4, 12, 2, 10, 6, 14, 1, 9, 5, 13, 3, 11, 7, 15}

index(A_i) = {0, 0, 0, 2, 0, 4, 4, 6, 0, 8, 8, 10, 8, 12, 12, 14}

b) pro takto specifická data

Z obrázku výše je možné vidět, že data jsou specificky uspořádaná, tj. pro ně můžeme navrhnout efektivnější algoritmus.

Ten nejdřív zjistí dvojkový logaritmus z velikosti pole. Poté prochází prvky, jejichž index je dělitelný 2^(log – 1), ale není dělitelný 2^log (aby se zabránilo přepisu). Prvkům, které toto splní, nastaví skok k nejbližšímu menšímu prvku na 2^(log – 1), odečte tento skok od indexu původního prvku a tím najde index nejbližšího menšího prvku.

Pseudokód:

```
mocnina = log2(velikostPole);
OPAKUJ (DOKUD mocnina >= 0){
        OPAKUJ (PRO index OD 0 DO velikostPole - 1){
                POKUD(index je dělitelný 2^(mocnina - 1) A není dělitelný 2^mocnina){
                        skok = 2^(mocnina - 1);
                        indexNejbližšíhoPrvku[i] = i – skok;
                }
                index = index + 1;
```

```
        }
        mocnina = mocnina - 1;
}
```

b) obecně

Pole se projde od prvku s indexem 1 do konce (nultý prvek má odkaz automaticky sám na sebe). Porovná se aktuální prvek s předcházejícím. Pokud je předcházející prvek menší, aktuální prvek si na něj (do vedlejšího pole) uloží odkaz. Pokud je předcházející prvek větší, aktuální prvek se podívá na odkaz menšího prvku. Pokud je i ten větší, aktuální prvek se podívá na odkaz ještě menšího prvku atd. Tímto se vyhneme porovnávání všech prvků každý s každým.

Pseudokód:

```
OPAKUJ (PRO index OD 1 DO velikostPole){
        POKUD (prvek na (indexu - 1) < prvek na aktuálním indexu){
                indexMenšího[index] = index – 1;
        }
        JINAK{
                porovnávanýIndex = index – 1;
                OPAKUJ (DOKUD prvek na aktuálním indexu < prvek na porovnávaném indexu){
                        porovnávanýIndex = indexMenšího[porovnávanýIndex]
                }
        }
}
```

2.

a)

$2^{n+1}=2\cdot 2^n\leq c\cdot 2^n\Rightarrow 2^{n+1}=O(2^n)$ (vyplývá z definice O)

b)

$2^{2n}=(2^2)^n=4^n\gg c\cdot 2^n\Rightarrow 2^{2n}\neq O(2^n)$ (vyplývá z definice O)

3.

a)

- $T(n)=9\cdot T(\frac{n}{10})+n$
- $a=9,\ \ b=10,\ \ f(n)=n;\quad \log_b a=\log_{10}9\approx 0{,}954$
- $f(n)=n=\Omega(n^{\log_{10}9+\epsilon})$

- $$a \cdot f(\frac{n}{b}) \leq c \cdot f(n);\quad n > n_0$$
  $$9 \cdot (\frac{n}{10}) \leq c \cdot n$$
  $$\frac{9}{c} \leq \frac{n}{(\frac{n}{10})}$$
  $$\frac{9}{c} \leq 10$$
- platí, protože $\frac{9}{c} > 9,\ c \in (0,1)$
- $\Rightarrow T(n) = \Theta(n)$

b)

- $T(n) = 16 \cdot T(\frac{n}{4}) + n^2$
- $a = 16,\ b = 4,\ f(n) = n^2;\quad \log_b a = \log_4 16 = 2$
- $f(n) = n^2 = \Theta(n^{\log_4 16})$
- $\Rightarrow T(n) = \Theta(n^2 \cdot \log n)$

c)

- $T(n) = 7 \cdot T(\frac{n}{3}) + n^2$
- $a = 7,\ b = 3,\ f(n) = n^2;\quad \log_b a = \log_3 7 \approx 1{,}771$
- $f(n) = n^2 = \Omega(n^{\log_3 7 + \epsilon})$
- $$a \cdot f(\frac{n}{b}) \leq c \cdot f(n);\quad n > n_0$$
  $$7 \cdot (\frac{n}{3}) \leq c \cdot n^2$$
  $$\frac{7}{c} \leq \frac{n}{(\frac{n}{3})^2}$$
  $$\frac{7}{c} \leq 9$$
- platí, protože $\frac{7}{c} > 7,\ c \in (0,1)$
- $\Rightarrow T(n) = \Theta(n^2)$

d)

- $T(n) = 7 \cdot T(\frac{n}{2}) + n^2$

- $a=7,\ \ b=2,\ \ f(n)=n^2;\quad \log_b a=\log_2 7\approx 2{,}807$
- $f(n)=n^2=\Theta(n^{\log_2 7-\epsilon})$
- $\Rightarrow T(n)=\Theta(n^{\log_2 7})$

e)

- nelze vyřešit obecně pomocí Master theorem
- $$\begin{gathered} T(n)=T(\sqrt{n})+1=T(\frac{n}{\sqrt{n}})+1 \\ (T(\frac{n}{\sqrt{\sqrt{n}}})+1)+1 \\ ((T(\frac{n}{\sqrt{\sqrt{\sqrt{n}}}})+1)+1)+1 \\ (\cdots(T(\frac{n}{\sqrt{\sqrt{\sqrt{n}}}\ddots})+\cdots+1)+1)+1 \end{gathered}$$
- $\log_{\sqrt{n}} n=2$
- $$\begin{gathered} 1^{\log_{\sqrt{n}} n}=n^{\log_{\sqrt{n}} 1} \\ 1^2=n^0 \\ 1=1 \end{gathered}$$
- $\Rightarrow T(n)=O(1)$

4.

$$\begin{gathered} 1 \ll n^{\frac{1}{\log n}}=2 \ll \log\log n \ll \sqrt{\log n} \ll \log n \ll \log^2 n \ll \\ \ll 2^{\sqrt{2\cdot\log n}}=n^{\sqrt{\frac{2}{\log n}}} \ll \sqrt{(2)}^{\log n}=\sqrt{n} \ll n \equiv 2^{\log n}=n \ll \log(n!) \ll n\cdot\log n \ll \\ n^2 \equiv 4^{\log n}=n^2 \ll n^3 \ll (\log n)! \ll n^{\log\log n} \equiv (\log n)^{\log n}=n^{\log\log n} \ll (\frac{3}{2})^n \ll 2^n \ll \\ \ll n\cdot 2^n=2^{n+\log n} \ll n! \ll (n+1)! \ll 2^{(2^n)} \ll (\log\log n)^{(2^n)} \end{gathered}$$

(vyřešeno hrubým seřazením podle vykreslení v Geogebře a následným porovnáním sousedních funkcí v nekonečnu ve Wolframu Alpha)