# ADS cvičení 2.5.2018

1. Popište třídící algoritmus, který bude vstup rozkládat na více než dvě části a ty pak rekurzivně třídit. Může být rychlejší než náš Mergesort?
2. Nalezněte nějaký algoritmus, který odpovídá druhému typu chování kuchařkové věty na řešení rekurencí (q<1).
3. Proč u quickselectu navrhujeme volit za pivota prvek xn/2 a ne třeba x1 nebo xn?
4. Jak by dopadlo, kdybychom na vstupu dostali posloupnost reálných čísel a jako pivota používali aritmetický průměr?
5. Máte zadáno v rovině n bodů. Najděte dvojici, která je nejblíže.
6. Inverze v posloupnosti říkáme každé dvojici (i, j) takové, že i<j a současně xi>xj. Vymyslete algoritmus, který spočítá, kolik daná posloupnost obsahuje inverzí. To může sloužit jako míra neuspořádanosti.

# ADS cvičení 2.5.2018

1. Popište třídící algoritmus, který bude vstup rozkládat na více než dvě části a ty pak rekurzivně třídit. Může být rychlejší než náš Mergesort?
2. Nalezněte nějaký algoritmus, který odpovídá druhému typu chování kuchařkové věty na řešení rekurencí (q<1).
3. Proč u quickselectu navrhujeme volit za pivota prvek xn/2 a ne třeba x1 nebo xn?
4. Jak by dopadlo, kdybychom na vstupu dostali posloupnost reálných čísel a jako pivota používali aritmetický průměr?
5. Máte zadáno v rovině n bodů. Najděte dvojici, která je nejblíže.
6. Inverze v posloupnosti říkáme každé dvojici (i, j) takové, že i<j a současně xi>xj. Vymyslete algoritmus, který spočítá, kolik daná posloupnost obsahuje inverzí. To může sloužit jako míra neuspořádanosti.

# ADS cvičení 2.5.2018

1. Popište třídící algoritmus, který bude vstup rozkládat na více než dvě části a ty pak rekurzivně třídit. Může být rychlejší než náš Mergesort?
2. Nalezněte nějaký algoritmus, který odpovídá druhému typu chování kuchařkové věty na řešení rekurencí (q<1).
3. Proč u quickselectu navrhujeme volit za pivota prvek xn/2 a ne třeba x1 nebo xn?
4. Jak by dopadlo, kdybychom na vstupu dostali posloupnost reálných čísel a jako pivota používali aritmetický průměr?
5. Máte zadáno v rovině n bodů. Najděte dvojici, která je nejblíže.
6. Inverze v posloupnosti říkáme každé dvojici (i, j) takové, že i<j a současně xi>xj. Vymyslete algoritmus, který spočítá, kolik daná posloupnost obsahuje inverzí. To může sloužit jako míra neuspořádanosti.