Dnes značí N velikost vstupu (zpravidla velikost nějakého pole) a B velikost bloku, který se přenáší mezi paměťmi. Hodnotu B zpravidla neznáme, ale pro analýzu jej můžeme používat. Často si ji tipnete na základě nějaké zkušenosti. Nějaká konstanta jako např $1 \ll 2^6 = 64$ je celkem ok.

1. Proveďte cache-oblivious analýzu následující implementace `QuickSelectu` (tj. algoritmus pro hledání mediánu).

   1. Pokud $N < \mathcal{O}(B)$, hledáme medián naivně.
   2. Vybereme uniformně náhodně pivota. Tuto volbu opakujeme, dokud neplatí, že vybraný pivot je v prostředních dvou kvartilech prvků v současném poli.
   3. Pole přeskládáme tak, že začne prvky menšími než pivot, pak následuje pivot a nakonec máme prvky větší než pivot.
   4. Pokud je pivot na indexu $N/2$, vrátíme jej jako medián.
   5. Pokud je pivot na indexu menším než $N/2$, rekurzivně hledáme medián v prvcích větším než pivot (a patřičně upravíme, kolikátý prvek hledáme).
   6. Pokud je pivot na indexu větším než $N/2$, rekurzivně hledáme medián v prvcích menším než pivot (a patřičně upravíme, kolikátý prvek hledáme).

   Nápověda 1. Počet přenesených paměťových bloků je nyní `n á h o d n á   v e l i č i n a`. Pro nás to dneska naštěstí jenom[1] znamená linearitu střední hodnoty. Pokud tedy $X = \alpha Y + \beta Z$ pro $\alpha, \beta \in \mathbb{R}$, pak $\mathbb{E}[X] = \alpha\mathbb{E}[Y] + \beta\mathbb{E}[Z]$.

   Nápověda 2. Je-li $X \sim \mathrm{Geom}(p)$, pak $\mathbb{E}[X] = 1/p$.

2. Proveďte cache-oblivious analýzu následující implementace `QuickSortu`.

   1. Pokud $N < \mathcal{O}(B)$, naivně vrátíme seřazené pole.
   2. Vybereme uniformně náhodně pivota. Tuto volbu opakujeme, dokud neplatí, že vybraný pivot je v prostředních dvou kvartilech prvků v současném poli.
   3. Pole přeskládáme tak, že začne prvky menšími než pivot, pak následuje pivot a nakonec máme prvky větší než pivot.
   4. Rekurzivně necháme seřadit pole před pivotem a pole za pivotem.

   Nápověda. Počet přenesených paměťových bloků je stále `n á h o d n á   v e l i č i n a`.

3. Proveďte cache-oblivious analýzu následujícího algoritmu pro hledání mediánu v poli A.

   1. Pokud $N < \mathcal{O}(B)$, vrátíme medián naivně.
   2. Rozdělme A na $\lceil n/5 \rceil$ souvislých úseků délky 5.
   3. Spočtěme medián v každé pětici pomocí $\mathcal{O}(1)$ operací.
   4. **Rekurzivně** spočtěme medián z těchto mediánů.
   5. Rozdělme A na dvě části podle toho, jestli je prvek větší či menší než rekurzivně spočtený medián.
   6. Spočtěme počet prvků v každé části z předchozího bodu a rekurzivně hledejme medián v části, kde by medián měl být.

   Nápověda 1. Můžete využít znalosti, že se v posledním kroku rekurzivně zanořujete do části velikosti nejvýše $\mathcal{O}\left(\frac{7}{10}N\right)$. (Nudíte-li se, dokažte tuto znalost)

   Nápověda 2. $\left(\frac{1}{5}\right)^x + \left(\frac{7}{10}\right)^x = 1$ má řešení $x \approx 0.8397803$.

[1] Doufejme.