

ALGORYTMY I STRUKTURY DANYCH

SORTOWANIE I WYBÓR

Instytut Informatyki Uniwersytetu Wrocławskiego

Paweł Rzechonek

1. [***] Rozważmy następujący pomysł na trójkpodział: z tablicy rozmiaru n losowo wybieramy dwa różne elementy; mniejszy z nich x przenosimy na początek tablicy a większy y na koniec; następnie przesuwamy pozostałe elementy w taki sposób, aby najpierw występowały $\leq x$ (pierwszy blok), potem te które są $> x$ i $\leq y$ (drugi blok), a na końcu $> y$ (trzeci blok). Napisz w pseudokodzie implementację takiego algorytmu (powinna to być funkcja, która po przedstawieniu wszystkich elementów, zwróci parę liczb: pozycje, od których rozpoczynają się drugi i trzeci blok). Jakie jest prawdopodobieństwo, że po takim trójkpodziale żaden z trzech bloków nie większy niż $\frac{1}{2}n$, jeśli każda para elementów dzielących mogła być wylosowana z jednakowym prawdopodobieństwem? Możesz założyć, że wszystkie elementy w tablicy są różne.
2. [***] Jak zmodyfikować algorytm *quick-sort*, aby w każdym przypadku głębokość wywołań rekurencyjnych była nie większa niż $\log n$ dla danych rozmiaru n ?
3. [***] Pokaż, jak zaimplementować algorytm *quick-sort*, aby działał on *w miejscu* (nie wolno korzystać z rekurencji).
4. [**] Mamy dostęp do algorytmu *czarna skrzynka*, który w liniowym czasie znajduje medianę nieuporządkowanego ciągu liczb. W jaki sposób, korzystając z *, wyznaczyć dowolny k -ty co do wielkości element w nieuporządkowanym ciągu liczb. Twój algorytm także powinien działać w czasie liniowym.*
5. [**] Przedstawiony poniżej algorytm dokonuje jednoczesnego wyboru minimum i maksimum spośród nieuporządkowanych elementów. Określ ile porównań będzie wykonywał ten algorytm dla danych rozmiaru n . Jak bardzo różni się ten wynik od dolnej granicy $\lceil \frac{3}{2}n - 2 \rceil$? Jak zachowuje się ten algorytm dla danych, których rozmiar jest potęgą 2?

```

function MINMAX-REC (key  $T[0 \dots n - 1]$ )  $\mapsto$  (key, key)
{
    if  $n = 1$  then return ( $T[0]$ ,  $T[0]$ );
    if  $n = 2$  then
        if  $T[0] \leq T[1]$  then return ( $T[0]$ ,  $T[1]$ ); else return ( $T[1]$ ,  $T[0]$ );
     $m \leftarrow \lfloor \frac{n}{2} \rfloor$ ;
     $(m_l, M_l) \leftarrow$  MINMAX-REC ( $T[0 \dots m - 1]$ );
     $(m_r, M_r) \leftarrow$  MINMAX-REC ( $T[m \dots n - 1]$ );
    return ( $\min\{m_l, m_r\}$ ,  $\max\{M_l, M_r\}$ );
}

```