

Zadanie 5.

Pokaż, że czas działania algorytmu Quicksort jest $\Theta(n \log n)$ gdy wszystkie elementy sortowanej tablicy są takie same.

(Quicksort z książki „Wprowadzenie do algorytmów”)

Quicksort(A, p, r)

```
1 if p < r
2   then q  $\leftarrow$  Partition(A, p, r)
3     Quicksort(A, p, q)
4     Quicksort(A, q+1, r)
```

Partition(A, p, r)

```
1 x  $\leftarrow$  A[p]
2 i  $\leftarrow$  p - 1
3 j  $\leftarrow$  r + 1
4 while True
5   do repeat j  $\leftarrow$  j - 1
6     until A[j]  $\leq$  x
7   repeat i  $\leftarrow$  i + 1
8     until A[i]  $\geq$  x
9   if i < j
10    then swap(A[i], A[j])
11    else return j
```

Jako, że wszystkie elementy tablicy są takie same, dla każdego p, r algorytm *Partition* będzie przebiegał tak samo – będzie zamieniał elementy pierwszy z ostatnim, drugi z przedostatnim itd., aż dojdzie do środka tablicy (mniej więcej, zależy od tego, czy n jest parzyste) i następnie zwraca jego indeks. Wtedy w procedurze *Quicksort* otrzymujemy dwa pod-problemy rozmiaru $n/2$. Koszt wykonania partycji wynosi $\Theta(n)$, więc złożoność czasowa przedstawia się następująco:

$$T(n) = 2T\left(\frac{n}{2}\right) + \theta(n)$$

Z Master Theorem otrzymujemy:

$$T(n) = \theta(n \log_2 n)$$