

Ćwiczenia 11

- 8.1 Poniżej przedstawiamy ulepszony algorytm BubbleSort sortujący niemalejąco n -elementową tablicę $a[1..n]$:

BubbleSort (revisited)

```

1:  $i := n + 1$ ;
2: repeat
3:    $i := i - 1$ ;
4:   posortowany := TRUE;
5:   for  $j := 2$  to  $i$  do
6:     if  $a[j] < a[j - 1]$  then
7:        $a[j - 1] := a[j]$ ;                                zamiana elementów
8:       posortowany := FALSE;
9: until posortowany;
```

Zaprojektuj efektywny algorytm, który dla tablicy a zawierającej permutację liczb $1, 2, \dots, n$ obliczy ile razy algorytm BubbleSort wykona pętlę **repeat**.

- 8.2 (a) Do początkowo pustego AVL-drzewa wstawiamy kolejno klucze $1, 2, \dots, n$, gdzie n jest dodatnią liczbą całkowitą. Podaj klucze, które kiedykolwiek znajdą się w korzeniach kolejno powstających drzew.
- (b) Jaka jest minimalna, a jaka maksymalna liczba węzłów (wewnętrznych) w AVL-drzewie o wysokości h ?
- (c) Załóżmy, że kluczami w AVL drzewie są kolejne liczby naturalne $1, 2, \dots$. Jaki najmniejszy, a jaki największy klucz może znaleźć się w korzeniu AVL-drzewa o wysokości h ?
- 8.3 Zaproponuj implementacje operacji Join i Split dla drzew typu "splay".
- 8.5 Zaproponuj implementacje operacji Join i Split dla 2-3-4 drzew.

Do samodzielnej pracy

- 8.4 Dokonaj analizy kosztu amortyzowanego operacji LocalSplay w przypadku 2a (patrzy wykład).