

# ALGORYTMY I STRUKTURY DANYCH - ćwiczenia

## I rok INFORMATYKA

### studia I stopnia

## Lista 5

1. Dla listy jednokierunkowej<sup>1</sup> podać algorytm, który realizuje:
  - a) dodanie elementu do listy przed zadany element,
  - b) dodanie elementu do listy po zadany elemencie,
  - c) usunięcie zadanego elementu z listy,
  - d) przeszukanie listy w celu znalezienia podanej wartości,
  - e) obliczenie ilości elementów listy,
2. Napisz procedurę zamieniającą miejscami elementy znajdujące się na pozycjach  $p$  i  $p \rightarrow \text{NEXT}$  listy jednokierunkowej.
3. Napisz procedurę łączenia dwóch posortowanych list w jedną posortowaną listę jednokierunkową.
4. Z danego stosu<sup>2</sup> liczb całkowitych zdejmując elementy utworzyć dwa stosy - jeden złożony z liczb parzystych, drugi z liczb nieparzystych.
5. Użyć stosu do odwrócenia kolejności liter tekstu.
6. Sprawdzić, czy na stosie są dwa identyczne elementy.
7. Sprawdzić, czy w danym fragmencie kodu programu nawiasy są wstawione poprawnie. Uwaga! Nie wystarczy sprawdzić, czy liczba nawiasów otwierających jest taka sama, jak zamykających. Np. Układ „{...{...}}” nie jest poprawny.
8. Dany jest stos  $S$  liczb całkowitych. Używając standardowych operacji na stosie podać algorytm *Stos uporządkowany*, który ustawia elementy na stosie  $S$  w porządku rosnącym (na szczycie jest element największy). Można korzystać z jednego pomocniczego stosu  $P$  i kilku (skończonej liczby) zmiennych.  
Przykład: (szczyt) 1,2,6,3,4,5  $\rightarrow$  (szczyt) 6,5,4,3,2,1.
9. Dane są: stos liczb  $A$  i pusty stos  $B$ . Używając standardowych funkcji obsługi stosów usunąć ze stosu  $A$  element o najmniejszej wartości położony najgłębiej (w stosie może być kilka elementów o tej samej najmniejszej wartości).
10. Dana jest tablica  $\text{int } A[N]$ . Podaj algorytm wyznaczania maksymalnego  $k$ , takiego że  $k = j - i$ ,  $0 \leq i \leq j \leq N-1$  oraz  $A[j] \geq A[i]$  (można to zrobić w czasie  $\Theta(N)$ )

<sup>1</sup> Przyjmujemy następującą definicję węzłów listy:

```
struct node {Item item; node *next;};  
typedef node *link;
```

<sup>2</sup> Definicja stosu:

```
STACK <Typ> nazwa;
```

dostępne metody:

```
Typ nazwa.POP();  
void nazwa.PUSH(Typ x);  
int nazwa.EMPTY(); // 1 - gdy stos pusty, 0 - gdy nie jest pusty
```

11. Dla kolejki<sup>3</sup> *K* podać algorytm, który realizuje:
- a) obliczenie ilości elementów,
  - b) sprawdzenie, czy zawiera element o danej wartości,
12. Jak zaimplementować kolejkę przy pomocy dwóch stosów? Jak zaimplementować stos przy pomocy dwóch kolejek?

---

<sup>3</sup> Definicja kolejki

QUEUE <Typ> nazwa

dostępne metody:

```
Typ nazwa.GET();           // DEQUEUE
void nazwa.PUT(Typ x);      // ENQUEUE
int nazwa.EMPTY();          // 1 - gdy kolejka pusta, 0 - gdy nie jest pusta
```