

Egzamin z ASD
4.03.2013

Zadanie 1 [10 punktów]

Dany jest początkowo pusty, dynamiczny ciąg elementów. Dozwolone jest dodawanie i usuwanie elementów na początku i na końcu ciągu. Zaproponuj algorytm symulujący te operacje z pomocą trzech stosów. Dokonaj analizy zamortyzowanego czasu wykonywanych operacji.

Zadanie 2 [10 punktów]

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

```
i := n+1;
do{
  i := i - 1;
  posortowany := TRUE;
  for j := 2 to i do
    if a[j] < a[j-1] then{
      a[j-1] := a[j]; // zamiana elementów
      posortowany := FALSE
    }
  }
until posortowany;
```

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

Wskazówka: spróbuj rozwiązać wektor inwersji dla permutacji zapisanej w tablicy a .

Zadanie 3 [10 punktów]

Grafy klikowe to grafy spójne, w których każda dwuspójna składowa jest kliką (grafem pełnym). Pamiętaj, że pojedyncza krawędź jest kliką dwuwierzchołkową. Grafy klikowe można reprezentować w następujący sposób:

- Przeglądamy graf metodą przeszukiwania w głąb, budujemy drzewo przeszukiwania i numerujemy wierzchołki $1, 2, \dots, n$ w kolejności odwiedzania. Wierzchołki utożsamiamy z ich numerami.
- Dla każdego wierzchołka $v > 1$ pamiętamy jego ojca $p[v]$ oraz wierzchołek $low[v]$ – najmniejszy wierzchołek (wierzchołek z najmniejszym numerem), do którego prowadzi co najwyżej jedna krawędź niedrzewowa od pewnego potomka v . (Uwaga: v jest swoim potomkiem.) Taką reprezentację nazywamy *oszczędną*.

Zaproponuj efektywne algorytmy, które mając daną oszczędną reprezentację grafu klikowego obliczą

- [5 punktów] z ilu dwuspójnych składowych składa się ten graf,
- [5 punktów] rozmiar najliczniejszego skojarzenia.

Zadanie 4 [10 punktów]

Zaproponuj algorytm dodania do n -elementowego drzewa lewicowego k kluczy w czasie $O(k + \log n)$.

Uwaga: w rozwiązaniu każdego zadania uzasadnij poprawność swojej odpowiedzi oraz dokonaj analizy złożoności obliczeniowej zaproponowanych algorytmów.