

Zadania powtórzeniowe z algorytmów

Zad.1 Dana jest poniższa tablica z elementami:

4	12	6	9	5	10	1	7	3
---	----	---	---	---	----	---	---	---

a) Przedstaw tablicę w formie kopca (drzewko) i zasymuluj na niej procedurę Build-Max-Heap.

„Kopiec” przed wywołaniem procedury	„Kopiec” po pierwszym Heapify	„Kopiec” po drugim Heapify
„Kopiec” po trzecim Heapify	Czwarte Heapify –po zamianie nr 1	Czwarte Heapify – po zamianie nr 2
Czwarte Heapify – po zamianie nr 3		

b) Dla powyższego stosu wywołaj procedurę Heapsort (krok w którym zbudowaliśmy stos Build-Max-Heap mamy oczywiście już zrobiony). Wystarczy, że posortujesz 4 elementy.

Kopiec po Build-Max-Heap	Kopiec po wyrzuceniu pierwszego elementu i wstawieniu na szczyt innego	Kopiec po naprawie za pomocą Heapify
Kopiec po wyrzuceniu drugiego elementu i wstawieniu na szczyt innego	Kopiec po naprawie za pomocą Heapify	Kopiec po wyrzuceniu trzeciego elementu i wstawieniu na szczyt innego
Kopiec po naprawie za pomocą Heapify	Kopiec po wyrzuceniu czwartego elementu i wstawieniu na szczyt innego	Kopiec po naprawie za pomocą Heapify

Zad.2 Dana jest poniższa tablica z elementami, którą chcemy posortować algorytmem Quicksort.

6	12	1	9	5	10	2	7	3
---	----	---	---	---	----	---	---	---

- a) Przedstaw jak zmieniać się będzie tablica w trakcie pierwszego wywołania procedury Partition. Oznaczaj pod komórkami tablicy zmieniające się indeksy i,j.

6	12	1	9	5	10	2	7	3
---	----	---	---	---	----	---	---	---

Przed sortowaniem. $x=.....$

--	--	--	--	--	--	--	--	--

Tablica po pierwszej zamianie (oznacz i,j)

--	--	--	--	--	--	--	--	--

Tablica po drugiej zamianie (oznacz i,j)

--	--	--	--	--	--	--	--	--

Tablica po trzeciej zamianie (oznacz i,j)

Zwrócony element $q=$

- b) Zasymuluj dalsze działanie procedury Quicksort (zwróć uwagę na kolejność wykonywania procedur Partition)

--	--	--	--	--	--	--	--	--

--	--	--	--	--	--	--	--	--

--	--	--	--	--	--	--	--	--

--	--	--	--	--	--	--	--	--

--	--	--	--	--	--	--	--	--

--	--	--	--	--	--	--	--	--

--	--	--	--	--	--	--	--	--

--	--	--	--	--	--	--	--	--

--	--	--	--	--	--	--	--	--

--	--	--	--	--	--	--	--	--

--	--	--	--	--	--	--	--	--

--	--	--	--	--	--	--	--	--

--	--	--	--	--	--	--	--	--

--	--	--	--	--	--	--	--	--

--	--	--	--	--	--	--	--	--

--	--	--	--	--	--	--	--	--

--	--	--	--	--	--	--	--	--

--	--	--	--	--	--	--	--	--

Zad. 3 Przeprowadź sortowanie metodą przez zliczanie (Counting-sort) dla tablicy A i $k=9$.

A:

2	1	9	5	3	2	7	3
1	2	3	4	5	6	7	8

a) Jak zmieniać się będzie zawartość tablicy pomocniczej C?

0	1	2	3	4	5	6	7	8	9

Pierwsza pętla (C po zerowaniu)

0	1	2	3	4	5	6	7	8	9

Druga pętla (C po zliczaniu)

0	1	2	3	4	5	6	7	8	9

Trzecia pętla (C po sumowaniu)

b) Jak będą wstawiane elementy do tablicy B, i jak zmieniać się będzie tablica C?

B:

1	2	3	4	5	6	7	8

Wstawiamy 1 element

C:

0	1	2	3	4	5	6	7	8	9

B:

1	2	3	4	5	6	7	8

Wstawiamy 2 element

C:

0	1	2	3	4	5	6	7	8	9

B:

1	2	3	4	5	6	7	8

Wstawiamy 3 element

C:

0	1	2	3	4	5	6	7	8	9

B:

1	2	3	4	5	6	7	8

Wstawiamy 4 element

C:

0	1	2	3	4	5	6	7	8	9

B:

1	2	3	4	5	6	7	8

Wstawiamy 5,6,7,8 element

C:

0	1	2	3	4	5	6	7	8	9

Zad 4. Wstaw do poniższej tablicy z haszowaniem T po kolei elementy metodą haszowania łańcuchowego z funkcją haszującą przez mnożenie ze stałą $A=0,6$.

Wstawiamy klucz 10:

0	
1	
2	
3	
4	
5	
6	

Wstawiamy klucz 13:

0	
1	
2	
3	
4	
5	
6	

Wstawiamy klucz 15:

0	
1	
2	
3	
4	
5	
6	

Wstawiamy klucz 3:

0	
1	
2	
3	
4	
5	
6	

Zad.5 Wykonaj operacje wstawienia lub usuwania na tablicy T z metodą haszowania otwartego. W przypadku kolizji zaznacz kolejne sprawdzane elementy. Elementy usuwane pozostawiają w komórce pewną stałą wartość **D** (jak Deleted).

a) Użyj adresowania liniowego z funkcją haszującą modularnie.

0	1	2	3	4	5	6	

Wstaw 8

0	1	2	3	4	5	6	

Wstaw 18

0	1	2	3	4	5	6	

Wstaw 15

0	1	2	3	4	5	6	

Wstaw 22

0	1	2	3	4	5	6	

Usuń 15

0	1	2	3	4	5	6	

Wstaw 29

b) Użyj adresowania kwadratowego z funkcją haszującą modularnie.

0	1	2	3	4	5	6	

Wstaw 8

0	1	2	3	4	5	6	

Wstaw 18

0	1	2	3	4	5	6	

Wstaw 15

0	1	2	3	4	5	6	

Wstaw 22

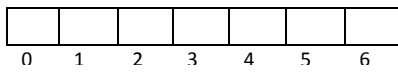
0	1	2	3	4	5	6	

Usuń 15

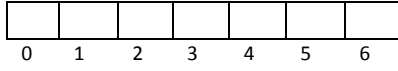
0	1	2	3	4	5	6	

Wstaw 29

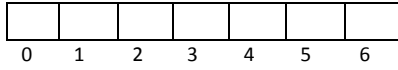
c) Użyj adresowania dwukrotnego z funkcjami h_1 i h_2 modularnymi.



Wstaw 8



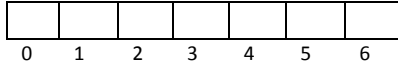
Wstaw 18



Wstaw 15



Wstaw 22



Usuń 15



Wstaw 29

Zad.6 Wykonaj na poniższym drzewie binarnym po kolei następujące operacje:

Usuń 23, Wstaw 1, Wstaw 4, Usuń 7, Usuń 30, Usuń 15

