Algorytmy i struktury danych

Przygotowanie do kolokwium

Przyjmując, że `t1[] = {1, 2, 3, 4, 5, 6, 7}` oraz `t2[] = {7, 6, 5, 4, 3, 2, 1}` i stosując algorytmy sortujące ściśle wg procedur z pliku `sorty2020.cc` i wykonaj polecenia:

Zadanie 1

Ile dokładnie porównań (między elementami tablicy) wykona `insertion_sort(t2)` a ile `insertion_sort(t1)`?

`insertion_sort(t1)`: $n - 1 = 6$ porównań
`insertion_sort(t2)`: $n - 1 + 15$ inwersji $= 21$ porównań

Zadanie 2

Ile co najwyżej porównań (między elementami tablic) wykona procedura scalająca `merge` dwie tablice n-elementowe?

$2n - 1$ porównań w przypadku gdy naprzemiennie w obu tablicach występują elementy rosnące

Zadanie 3

Jaka jest pesymistyczna złożoność czasowa procedury `merge_sort`? Odpowiedź uzasadnij.

$$T(n) = 2T(n/2) + O(n)$$
$$T(n) = O(n \log n)$$

Zadanie 4

Ile co najwyżej porównań (między elementami tablicy) wykona procedura `partition`?

ilość porównań $\leq n + 1$

Zadanie 5

Jaka jest średnia a jaka pesymistyczna złożoność `quick_sort`. Odpowiedź uzasadnij.

Średnia: $T(n) = 2T(n/2) + n = O(n \log n)$

Pesymistyczna: $T(n) = T(n - 1) + n + 1$ z sumy ciągu arytmetycznego $O(n^2)$

Zadanie 6

Jaka jest złożoność funkcji `buildheap`? Przeprowadź dowód - uzasadnij swoją odpowiedź.

$$2\sum_{i=1}^{h-1} \frac{n}{2^{i+1}} \cdot i$$

$$2(\frac{n}{4} \cdot 1 + \frac{n}{8} \cdot 2 + \frac{n}{16} \cdot 3 + \frac{n}{32} \cdot 4 + \dots)$$

$$\frac{n}{2}(\frac{1}{1} + \frac{2}{2} + \frac{3}{4} + \frac{4}{8} + \dots)$$

$$\frac{1}{1} + \frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \dots = \frac{\frac{1}{1}}{1 - \frac{1}{2}} = 2$$

$$\frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \dots = 1$$

$$\frac{1}{4} + \frac{1}{8} + \dots = 0.5$$

$$\frac{1}{8} + \dots = 0.25$$

$$\dots$$

$$\frac{n}{2} \cdot \frac{2}{1 - \frac{1}{2}} = \frac{n}{2} \cdot 4 = 2n = O(n)$$

Zadanie 7

Ile dodatkowej pamięci wymaga posortowanie tablicy n-elementowej za pomocą algorytmu: (a) `mergesort` (b) `quicksort` (c) `heapsort` (d) `insertionsort` (e) `countingsort` (f) `bucketsort` (g) `radixsort`. W punktach (e), (f), (g) zakładamy, że ilość kubełków jest m, a liczby do posortowania mają nie więcej niż k cyfr.

Algorytm	**Pamięć**
`mergesort`	$O(n)$
`quicksort`	$O(n)$
`heapsort`	$O(1)$
`insertionsort`	$O(1)$
`countingsort`	$O(n+k)$
`bucketsort`	$O(n+m)$
`radixsort`	$O(n+r)$

Zadanie 8

Jaka jest średnia a jaka pesymistyczna złożoność czasowa algorytmu: (a) `mergesort` (b) `quicksort` (c) `heapsort` (d) `insertionsort` (e) `countingsort` (f) `bucketsort` (g) `radixsort`? Zakładamy oznaczenia z poprzedniego zadania.

Algorytm	**Średnia**	**Pesymistyczna**
`mergesort`	$O(n \log n)$	$O(n \log n)$
`quicksort`	$O(n \log n)$	$O(n^2)$
`heapsort`	$O(n \log n)$	$O(n \log n)$
`insertionsort`	$O(n^2)$	$O(n^2)$
`countingsort`	$O(n+m)$	$O(n+m)$
`bucketsort`	$O(n+m)$	$O(n^2)$
`radixsort`	$O(nk)$	$O(nk)$

Zadanie 9

Udowodnij, że wysokość (ilość poziomów na których występują węzły) kopca n-elementowego wynosi $\lfloor \log_2 n \rfloor + 1$.

Maksymalna ilość węzłów w kopcu o wysokości h:

$$n(h) = 2^h - 1$$

$$n(h-1) = 2^{h-1} - 1$$

Minimalna ilość węzłów w kopcu o wysokości h:

$$n(h-1) + 1 = 2^{h-1} - 1 + 1 = 2^{h-1}$$

Ilość węzłów w kopcu o wysokości h:

$$2^{h-1} \leq n < 2^h$$

$$h - 1 \leq \log_2 n < h$$

$$h \leq \log_2 n + 1 < h + 1$$

$$\lfloor \log_2 n \rfloor + 1$$

Zadanie 10

Który element tablicy `t` jest (a) lewym dzieckiem (b) prawym dzieckim (c) ojcem, elementu `t[i]` w procedurze `heapsort`?

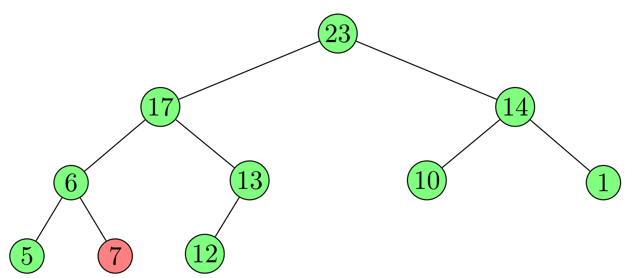
(a) `t[2i]`

(b) `t[2i+1]`

(c) `t[(i - 1)/2]`

Zadanie 11

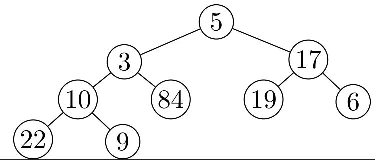
Czy ciąg `{23,17,14,6,13,10,1,5,7,12}` jest kopcem?



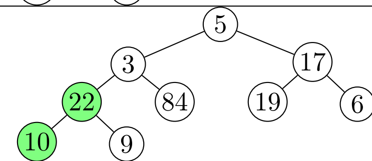
Zadanie 12

Zilustruj działanie procedury buildheap dla ciągu `{5,3,17,10,84,19,6,22,9}`. Narysuj na kartce wygląd tablicy/kopca po każdym wywołaniu procedury `przesiej`.

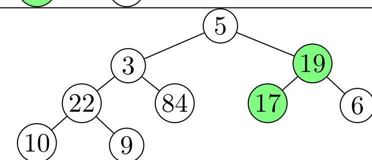
`{5,3,17,10,84,19,6,22,9}`



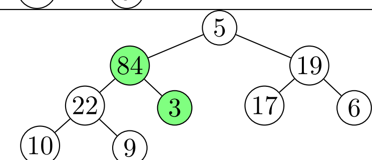
`{5,3,17,22,84,19,6,10,9}`



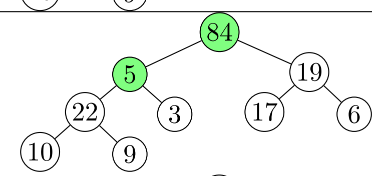
`{5,3,19,22,84,17,6,10,9}`



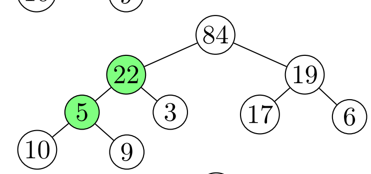
`{5,84,19,22,3,17,6,10,9}`



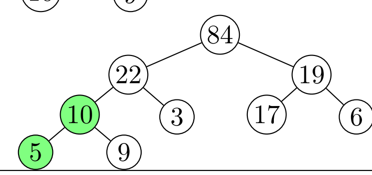
`{84,5,19,22,3,17,6,10,9}`



`{84,22,19,5,3,17,6,10,9}`



`{84,22,19,10,3,17,6,5,9}`



Zadanie 13

Zasymuluj działanie polifazowego mergesorta dla tablicy `{9, 22, 6, 19, 14, 10, 17, 3, 5}`. Na każdym etapie sortowania scala się sąsiadujące listy rosnące.

```
{9,22|6,19,14|10,17|3,5}
{6,9,14,19,22|3,5,10,17}
{3,5,6,9,10,14,17,19,22}
```

Zadanie 14

Zasymuluj działanie `mergesort(t2)`.

Rekurencyjny podział	**Merge**
`{7,6,5,4,3,2,1}`	`{7|6|5|4|3|2|1}`
`{7,6,5|4,3,2,1}`	`{7|5,6|3,4|1,2}`
`{7|6,5|4,3|2,1}`	`{5,6,7|1,2,3,4}`
`{7|6|5|4|3|2|1}`	`{1,2,3,4,5,6,7}`

Zadanie 15

Zasymuluj działanie `partition(t2, 7)`.

`pivot = t2[7 / 2] = t[3] = 4`

Tablica	**k**	**n**
`{7,6,5,4,3,2,1}`	-1	7
`{1,6,5,4,3,2,7}`	0	6
`{1,2,5,4,3,6,7}`	1	5
`{1,2,3,4,5,6,7}`	2	4
`{1,2,3,4,5,6,7}`	3	3

`return 3`

Zadanie 16

Zasymuluj działanie `partition(t2, 7)` w przypadku gdyby piwotem zamiast `t[n/2]` było `t[0]`.

`pivot = t2[0] = 7`

Tablica	**k**	**n**
`{7,6,5,4,3,2,1}`	-1	7
`{1,6,5,4,3,2,7}`	0	6
`{1,6,5,4,3,2,7}`	6	5

`return 6`

Zadanie 17

Wykaż, że pesymistyczna złożoność `quicksort` wynosi $O(n^2)$.

$$T(n) = T(n-1) + n + 1 \implies \sum_{i=1}^{n} i + 1 = \frac{n(2+n+1)}{2} = O(n^2)$$

Zadanie 18

Napisz wzór na numer kubełka, do którego należy wrzucić liczbę x w sortowaniu kubełkowym, jeśli kubełków jest n, a elementy tablicy mieszczą się przedziale (a, b). Numeracja zaczyna się od 0.

$$k = \left\lfloor \frac{x-a}{b-a} \cdot n \right\rfloor$$

Zadanie 19

Jak obliczyć k-tą od końca cyfrę w liczby x? Jak obliczyć ilość cyfr liczby x? Przyjmujemy układ dziesiętny. Jak wyniki zmienią się w układzie pozycyjnym o 1000 cyfr?

$$x_k = \left\lfloor \frac{x}{m^k} \right\rfloor \mod m \qquad n = \lceil \log_m x \rceil$$

$$x_k = \left\lfloor \frac{x}{10^k} \right\rfloor \mod 10 \qquad n = \lceil \log_{10} x \rceil$$

$$x_k = \left\lfloor \frac{x}{1000^k} \right\rfloor \mod 1000 \qquad n = \lceil \log_{1000} x \rceil$$