

2,5	4	4	4	2,5	17	db
-----	---	---	---	-----	----	----

HS

b) Złożoność HS'a dla wszystkich przefch
 Przypadek wynosi: $O(n \log n)$. Budowanie oraz
 przwracanie właściwości stogu zatraca $O(n)$
 czasu, natomiast procedura HEAPIFY, która
 przechodzi przez całą wysokość przewa kosztuje
 $O(\log n)$
 * zamiana korzenia głównego z
 ostatnim liściem

Złożoność OS'a to dla przypadku średniego oraz najlepszego $O(n \log n)$.
 Odpowiednio zastosowana wykorzystana procedura
 DZIEL i ZWYCIĘZA przedstawia się następująco
 DZIEL: działa w czasie ~~$O(n)$~~ stałym
 ZWYCIĘZA: sortowanie podtablicy $O(\log n)$

$$\text{SCALAJ: } O(n) \cdot T(n) = L(T/2) + O(n)$$

wynikająca rekurencji
 podział podzieli tablicę dającą
 w postacie

UWAGA! Czy mamy do czynienia
 z tablicą posortowaną lub odwrotnie
 posortowaną? Wtedy złożoność generuje
 się do $O(n^2)$ i przypomina bardziej
 SELECTION SORT'A

+23265

KILO

wybierając zawsze
 * ORAZ z piwotem na koncu STR 1

1 2 3 4 5 $\xrightarrow{\text{pivot}} (n-1)$

\downarrow
~~(n-1)~~ 1 2 3 ~~4~~ $\xrightarrow{\text{pivot}} (n-2)$

\downarrow
1 2 3 ~~4~~ $\xrightarrow{\text{pivot}} (n-3)$

\downarrow
itd.

REFERENCJA

UNIWERSALNA ZDECENEGROWANIE

SIE DO PRZEHODZENIA PRZECIĘ
WSEJSZKIE $n-1$ elementów

$$T(n) = T(n-1) \dots + O(n)$$

~~a) b)~~ c) SORTEWANIE W MIEJSCU

NASTĘPUJE CDT ~~ZAMIENIANIE ELE~~

SORTOWANE ELEMENTY ZAMIENIANE

SA, W TABLICZ WEJSCIOWEJ

BEZ KONIECZNOŚCI WYZKRYTYWANIA

POMOCNICZEJ

STRUKTURY DANYCH

np.
K, SS, HS
insertion, selection, heap sort

STR G
STR J

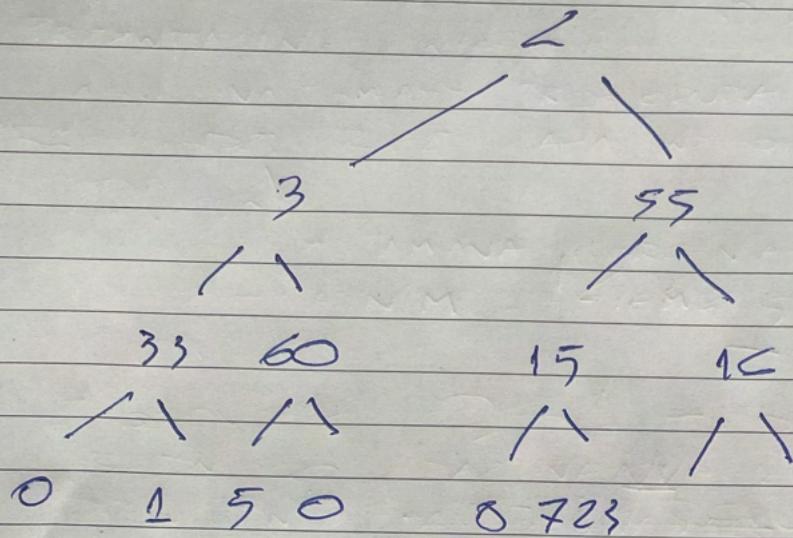
JAKUB KNITER

129528

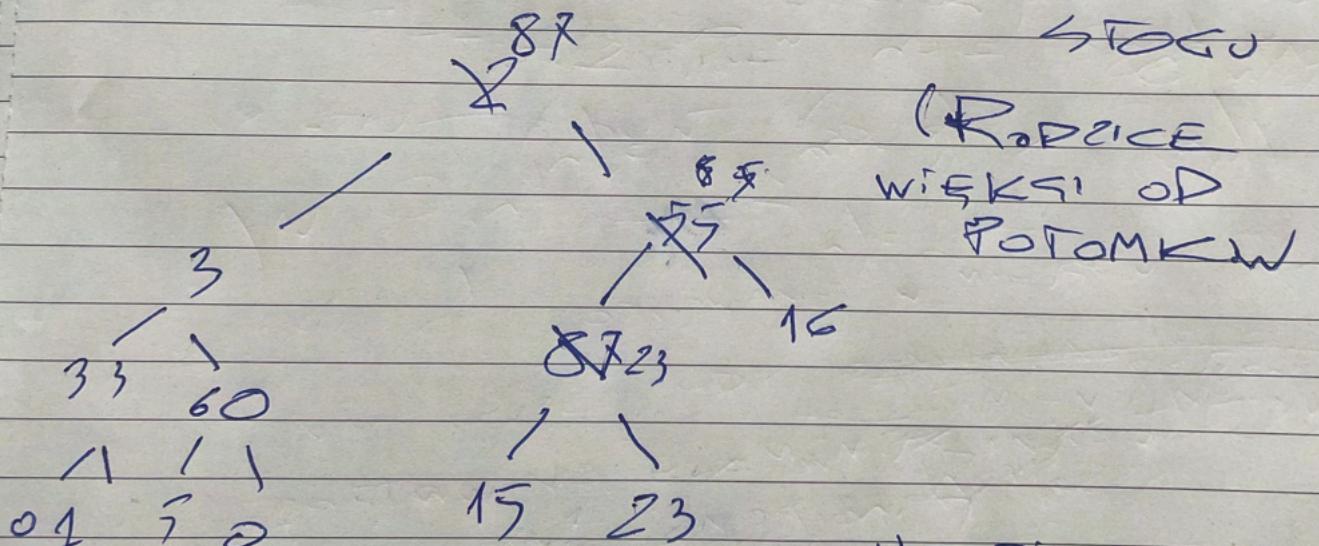
19.06.2021

2, 3, 55, 33, 60, 15, 16, 0, 1, 5, 0, 87, 23
1

BUDOWANIE
DRZEWA



PRZYWRACANIE
WEWNĘTRZ



STOGI
(Rodzice
więksi od
potomków)

HEAPIFY
ZAMIANA PIERWSZEGO
KOWAN

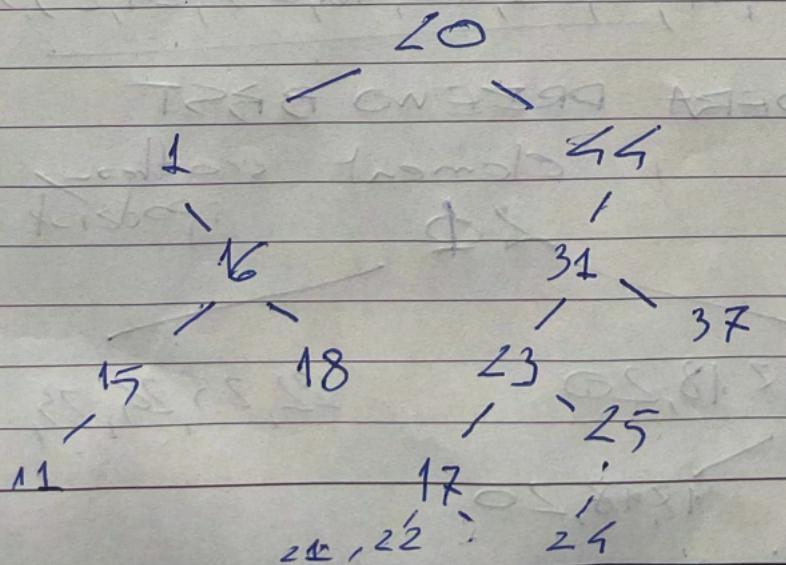
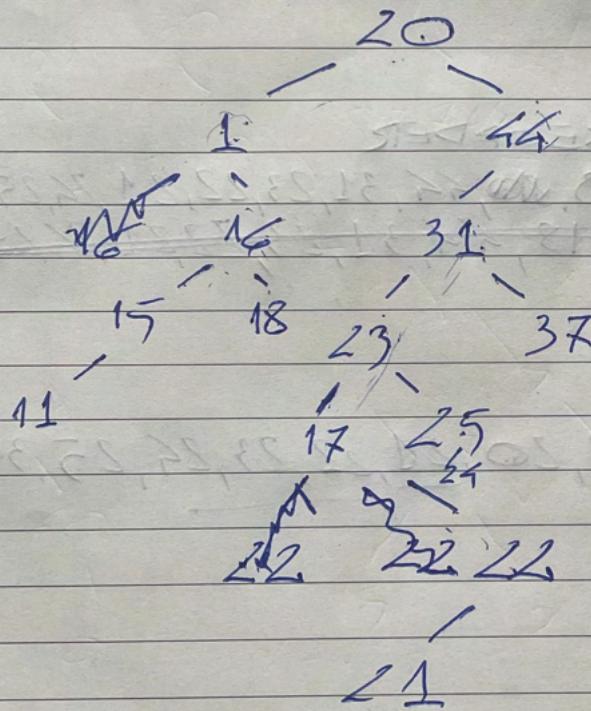
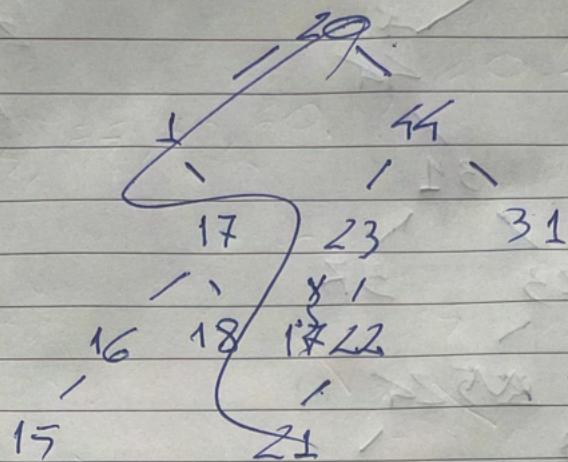
KILO

STR 3

JAKUB KNITTER

121528

19.06.2021

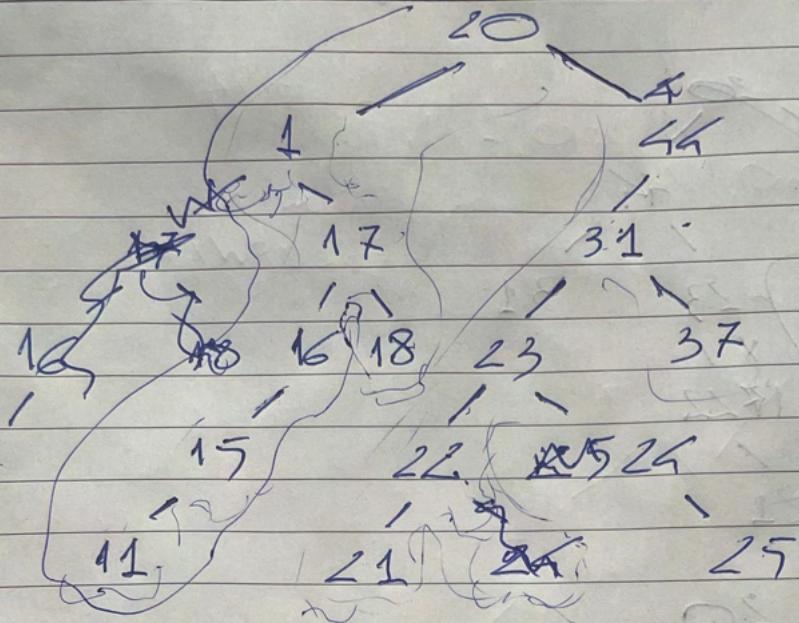


20, 44, 31, 1, 23, 17

22, 15, 18, 17

21, 77, 25

11, 24



Przeglądarka PREORDER

20, 1, 17, 16, 15, 11, 18, 44, 31, 23, 22, 21, 26, 25, 37, 44
~~20, 1, 17, 16, 15, 11, 18, 44, 31, 23, 22, 21, 26, 15, 37~~

IN ORDER

1, 11, 15, 16, 17, 18, 20, 21, 22, 23, 24, 25, 31, 37, 44

POST ORDER

11, 15, 16, 18, 17, 1, 21, 22, 25, 26, 23, 37, 31, 44, 20

z p. IN ORDERA DRZEWO BST
element środkowy podział

24

1, 11, 15, 16, 17, 18, 20

22, 25, 26, 23,

1, 11, 15 itd. 17, 18, 20

CTR G

JAKUB KNITTER

19518

19.05.2021

d)

① USUNAWIE I ZTARIASZA
najmniejszego elementu z
jednego podzcawu z
elementem sąsiednim

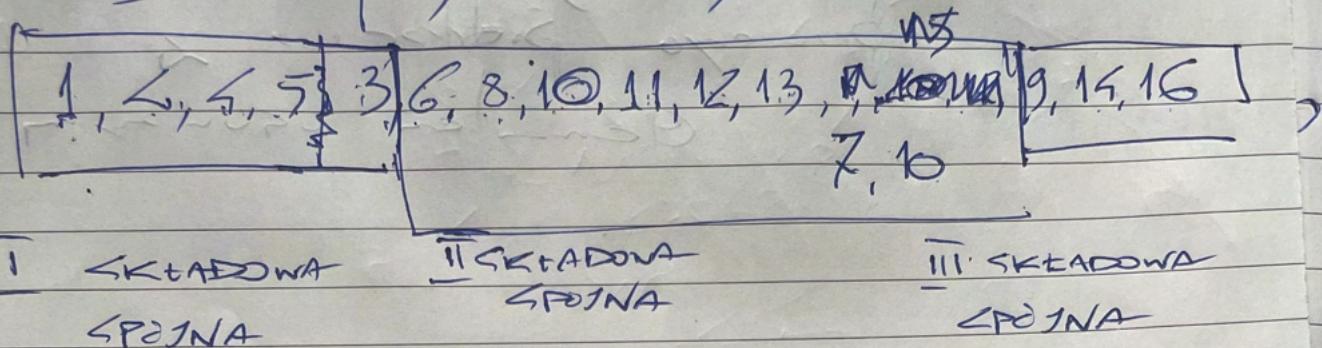
ZULU

GTR R

ZADANIE 3

a

DFS - preglödning inför



BFS user

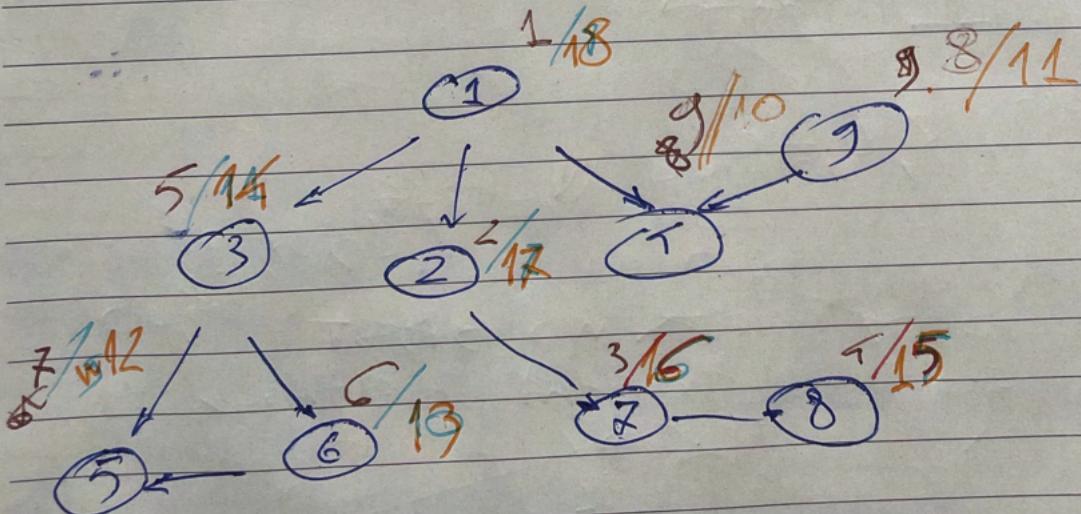
1, 2, 3, 5, 6, 8, 11, 12, 13, 15, 10, 7, 9, 15, 16

۶۱

2 forms of oblique views

naive oblivious methods, students, $O(n^2)$

metoda przedstawiajsząca porządek $O(n+m)$



~~18, 17, 16,~~

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

TANGO

ZADANIE 4

DLA SITUACJI W KTÓREJ ROZMIAR ELEMENTU

a) f DLA SITUACJI W KTÓREJ ROZMIAR ELEMENTU
 JEST WIĘKSZY OD ROZPATRYWANEGO
 ROZMIARU PLECAKA

$$\varsigma(a_i) > l$$

$$f(i, l) = f(i-1, l)$$

$w \nrightarrow (a_{i-1})$

PRZECIWNA WYPADKU

$$\varsigma(a_i) \leq l$$

$$f(i, l) = \max \left\{ \underbrace{f(i-1, l - \varsigma(a_i) + w(a_i))}_{\text{LICZBA "skoków"}}, f(i-1, l) \right\}$$

i	w _i	ς_i	0	1	2	3	4	5	6	7
1	3	4	0	0	0	0	3	3	3	3
2	4	2	0	0	2	2	2	2	5	5
3	6	3	0	0	2	6	6	8	8	9
4	12	1	0	12	12	14	18	18	20	20
5	19	2	0	12	14	26	26	28	32	32
6	44	16								

d) ALGORYTM PROGRAMOWANIA
JEN

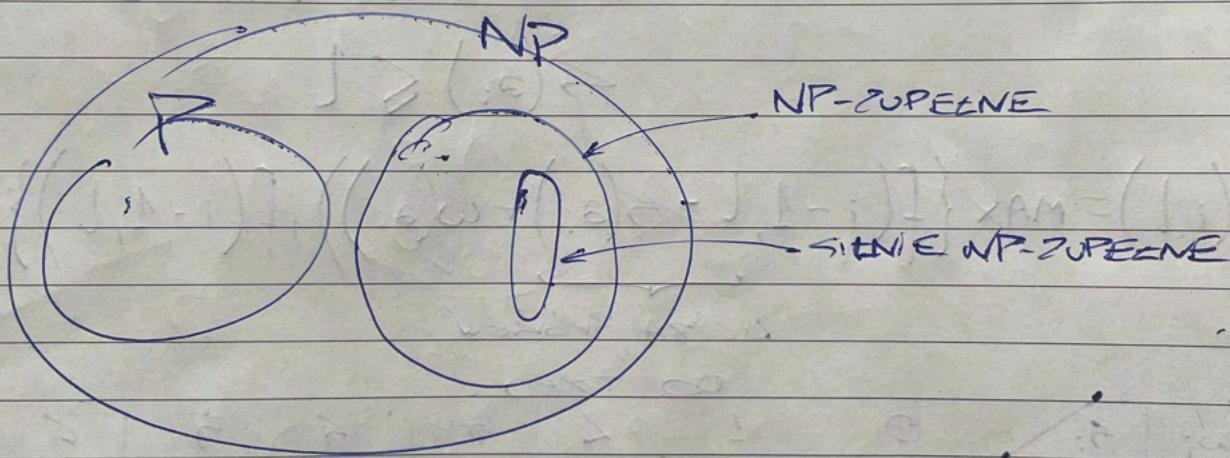
PONIJAŁA ZDZIĘCZNOŚĆ PSEUDOWIECOMIANAWY

(i, l - indeksy od siebie) wynosząca, w notacji O

$$\mathcal{O}(n^m)$$

c) kiedy podstawa podawana w PET A
PROBLEMU DA TAKIE ROZWIAZANIE
OPTIMALNE

ZADANIE 5



Aby udowodnić NP-ZUPELNOŚĆ DANEGO PROBLEMU NALEŻY
SPŁEĆ DWA WARUNKI

- 1) UDOWODNIĆ, ŻE PROBLEM JEST ~~NP-TRUDNY~~ NALEŻY DO NP.
- 2) ~~ZARZĄDZAĆ ZNAMY / PRZETRANIKOW~~
UDOWODNIĆ, ŻE PROBLEM NALEŻY DO KLASTY
PROBLEMÓW NP-ZUPEŁNIECH

Pierwszy punkt reakcja temu poprzez ~~więc~~
np. ~~wygenerowanie~~ utworzenie ~~wielomianów~~
instancji rozwiązania problemu oraz
weryfikację fizyczne ~~z danymi~~ dla ~~nie~~ własnych
rozwiązań
(rozwiązywanie w czasie
wielomianowym)

* za pomocą zadawanego algorytmu

Druż punkt to wybór znanego problemu
np. zupełnego (w tym przyp. dobrze wiadomo
żeż odrzucenie kolorowania grafu lub problemu
podziału podzbioru) i tego a następnie
tego redukcji do zadawanego problemu (P_{known})

Jesli taki redukcji następuje w czasie wielomianowym
(wszystkie problemy np.-zupełne transformują
się po siebie wzajemnie wielomianowo),
oznacza to, że problem będzie NP-zupełny

uzasadnieniu, że problem jest ~~NP~~
A więc w punkcie pierwszym najpierw zidentyfikować
algorytm, który dla pewnego podzbioru ~~zadawanego~~
~~zadawanego~~ pewnego rozwiązania dawałby poprawny (ale niegdyś)
wynik w czasie wielomianowym

Brawo