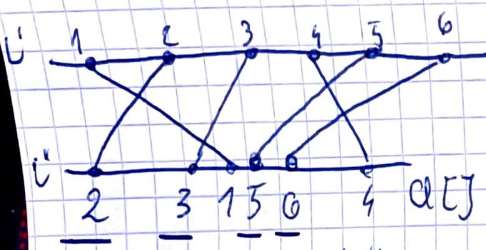
zad.8 a)



• Zauważmy, że ~~każdy~~ tylko podciąg rosnący a[] jest poprawnym zbiorem.

(⇒) $i < j \Rightarrow a[i] < a[j]$, rzeczywiście linie się nie przetną

(⇐) Jeżeli linie się nie przecinają to dla każdych: $i < j$, $a[i] < a[j]$ w.p.p. się przetną

• Szukamy więc najdłuższego podciągu rosnącego w a[]

d[i] - minimalny element kończący pewnego LIS'a odł. i

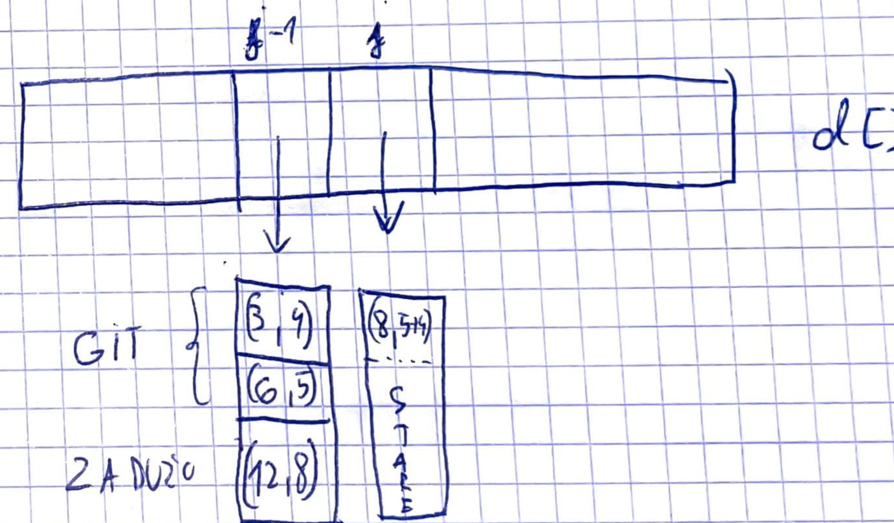
```
d[0] = -INF , d[0].idx = -1
d[1:n] = INF

for i in range (1,n):
    j = upper_bound(1,n,a[i]) // pierwszy większy w d[]
    d[j].idx = i
    d[j] = a[i]
    p[i] = d[j-1].idx

pos = d[k].idx // k - ostatni nie INF
OPT = []
WHILE pos != -1:
    OPT.APPEND(a[pos])
    pos = p[pos]
```

ODZYSKIWANIE WYNIKU

8. w)

- Aby znaleźć liczbę LIS'ów musimy trzymać jeszcze historię ich.
- Dla podciągu każdej długości będziemy trzymać wektor zawierający końcówki i ich krotności

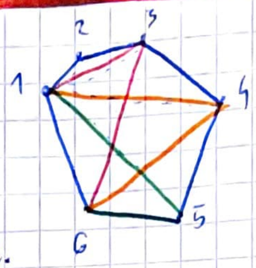
DODAJEMY 8



MODYFIKACJE:

- upperbound jest po d[1:n][0]
- UPD8 jak na obrazku

zad. 9



- Zauważmy, że bok 1-n w każdej triangulacji będzie bokiem pewnego trójkąta.
- Możemy więc „~~po~~ iterować" się po wierzchołkach $k \in [2, n-1]$ i szukać rozwiązania t.ż jest w nim trójkąt $\triangle$ (wierzchołki 1, n, k). Oznaczmy $T_{i,j}$ – minimalna ~~najdł~~ najdłuższa przekątna w triangulacji od i do j, dla dowolnego $k \in [2, n-1]$ mamy:

$$T_{1,n} = \max\{T_{1,k}, T_{k,n}, |P_1, P_k|, |P_n, P_k|\}$$ ← każde rozwiązanie tej jest dla pewnego k

więc OPT:

$$T_{1,n} = \min_{2<k<n} \max\{\overbrace{T_{1,k}, T_{k,n}}^{\text{PODPROBLEMY}}, |P_1, P_k|, |P_n, P_k|\}$$

Aby policzyć $T_{1,n}$ musimy wyliczyć $O(n^2)$ mniejszych wartości.

$$T_{i,j} = \begin{cases} 0 & i=j \vee i+1=j \vee i+2=j \\ \min\limits_{i<k<j} \max\{T_{i,k}, T_{k,j}, |P_i, P_k|, |P_j, P_k|\} \end{cases}$$

Stąd: $O(n^3)$

Aby odzyskać wynik możemy trzymać dodatkową tablicę $D[i][j]$ w której trzymamy wierzchołek, którym dzielimy wielokąt aby otrzymać opta.
Aby odzyskać wynik „idziemy" po tablicy:

$D_{1,n} = k_1 \longrightarrow$ dodać $(1, k_1), (n, k_1) \longrightarrow$ $D_{1,k_1} = k_2$ dodać $(1, k_2), (k_1, k_2)$; $D_{k_1,n} = k_3$ dodać $(k_1, k_3), (k_3, n)$