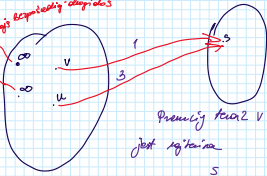
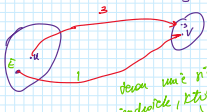


Dijkstra

bo nie majsi asymptotycznie



Przebieg przez V do S to najlepszy
jako optymalny



decyduje się o tym, że jest
winnym, który ma od niego do V
i koszt $E \rightarrow V \rightarrow S < \text{nie } V \rightarrow S$

zatem takie zachowanie minimum \Rightarrow kolejność priorytetowa
potrzebujemy operacji decrease-key

wstawiamy kopie potrzebujemy w naszym przypadku logn

bo może po zmniejszeniu klucza trzeba by
było przepisać wszystkie a do końca

operacja decrease-key
kalkulacja dodatkowa

minimum przedkładać kłopoty
można do minimum min.
czy właśnie jest to



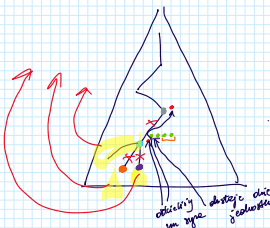
zorientować do góry
obcinamy kłopoty (zmniejszamy)
z całym jego podzestawą
i nie musimy nie listy
i nie musimy nie listy
(o ile po zmniejszeniu klucza nie ma
z nowym jest to niepotrzebne - npp)
nie ma obcinamy
znowe w tym, że trzeba już jest to
w przypadku gdy jest to drugi npp, to obcinamy
oprócz niego też ogra

jednostki kredytowe operacji dla kopci Fibonacciego:

- insert - 2, bo 1 dla końca a drugi po operacji niedzieli się na dwa
- min - 1, bo odczytanie
- delete min - $O(\log n)$
- decrease key - 4 - 1 jednostka na operację nowego min.
1 jednostka na koniec końca odczytania
2 jednostki na koniec ogra

NIEZMIENNIKI

- w końcu jest 1 jednostka
- na koniec niechciało nam, który
chodził z tego samego - 2 jednostki



tena jak obcinamy kłopoty (pomocny)

do jego kłopoty i potrzebujemy
jednostki, ale mamy takie odczytanie
ale mamy w tym 4 jednostki
kiedyś, nawet mamy to zrobić

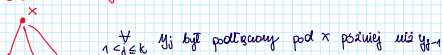
\Rightarrow PROBLEM 2. DELETE MIN - tena nie musimy zbliżyć
jak w kopiach eliminacji kłopoty, bo
coś od nich musimy odczytać

operacja join \Rightarrow na kopciach fibonacciego będzie działała trochę inaczej niż eliminacja

eliminacja będzie działać inaczej. o tym samym może być coś, że nie ma, tylko samo się podzieli pod koniec

W kopciach fibonacciego dwa $O(\log n)$ może być logarytmicznie wiele

Rozważmy dowolny moment w kopciach fibonacciego (po operacjach eliminacji, wstawiania itp.)
i dowolny węzełek x. Załóżmy, że x ma k synów



$1 \leq i \leq k$ to być podzielił pod x później niż y_i

same delete (n,p)

możemy więc użyć decrease key $O(n)$
kłopot na -2
z tegoż obcinamy
delete min $\Rightarrow O(\log n)$

Dej kłopot

kopci fibonacciego
to to może

i dowolny nienulowy x . Założymy, że x ma k symboli



cofamy się do momentu, gdy y był podziwiany pod x .

Je spin σ tym momentem x ? Co najmniej $i-1$ (bo teraz nie ma już i więc od 1 do $i-1$ musimy)
 Jaki mał. moment y_i ? co najmniej $i-1$ (bo mał. x to był co najmniej $i-1$ zastawo unieść)

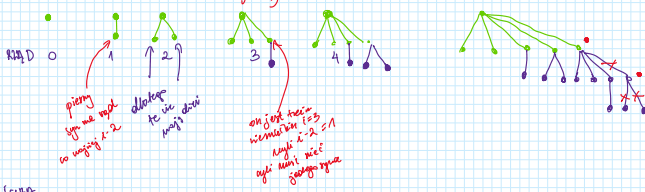
Wracamy do dowolnego momentu

Alle symboles me y_i observe (w dandyn marie)?

My: mości urzucić jednog syna, godyby stać dwiła do
byłby odciży, a niemy, ie jek wo usie, wten
mógł stać tyko jednog syna.

niektóre dane nie są możliwe i-2 tygodni, zatem
nie są i-2

Kaupujący płaci o medycie, minimalnie pod względem liczby mieszkańców
fibonacci $k_n = F_n$



Liuba
MERSCHAKEN 1 2 3 5 8 13

Nuotik: minimalus dreno o stepin i ma F_{i-2} minimalus

$$2 \left(\frac{1 + \sqrt{5}}{2} \right)^i = \phi$$

Cyfrę krotkę nazywamy o krotką, jeżeli nie istnieje n i ϕ^n
cyfry $k \leq O(\log k)$

mykłość (ogrzewanie przedlogu), bo

matrykałem stopniem konieczne to login, etc.

maximaln stopni najvišandij stepa nije to $\log n - 1$,

2. Zwiększenie wydajności i skrócenie czasu dobiegu, czyli

Take logn say scheduling