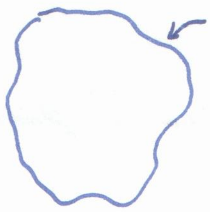
ALGORYTMY ZACHŁANNE



zbiór obiektów = C

ROZW: $S \subseteq C$

CEL: Koszt(S) – najmniejszy

Zaczynamy od S. i robimy $S \leftarrow S \cup \{e\}$, gdzie e jakoś wybieramy.
Z reguły e wybieramy nie zawsze: bo tak się nam wydaje, że będzie ok
(UWAGA, bo to nie zawsze ofc działa)

PRZYKŁAD:

Dane: $c_1, c_2, \dots, c_n \in \mathbb{N}$, $k \in \mathbb{N}$

Rozw: znaleźć $c_{i_1}, c_{i_2}, c_{i_3}, \dots c_{i_r}$, t.że $\sum_{i=1}^{r} c_{i_i} = k$

Cel: r – minimalne

Wydaje się, że dobre wziąć ~~...~~ największy możliwy nominał, który się mieści.

Przykład: $\{1,1,2,2,2,5,5,10,10\}$, $k=27$ MONETY

$S=\{10,10,5,2\}$

$\{1,2,5,10\}$ $k=27$ NOMINAŁY (nieskończ. wiele monet każdego rodzaju)

Czy ta strategia zachłanna jest dobra?

nie znajduje rozw. / znajduje jakieś, ale nie najlepsze

$C=\{3,5,9,10\}$ $k=14$ $\longrightarrow$ strategia zachłanna nie znajduje rozwiązania, chociaż istnieje, a dla $k=18$ będzie nieoptymalne

Pytanie dla jakich C ta strategia zachłanna daje rozw. optymalne?

MST

$G=(V,E,c) \rightsquigarrow T=(V,E',c)$, $\sum c$ – minimalne

1.) Kruskal

bez nieprawdy $c(e_i)$

1. Niech $e_1 \leqslant e_2 \leqslant \dots \leqslant e_n$
2. $E' \leftarrow \emptyset$
3. dla $i = 1 \dots m$
   if $E' \cup \{e\}$ nie zawiera cyklu
   then $E' \leftarrow E' \cup \{e_i\}$

2.) Prima

1. $E' \leftarrow \emptyset$
2. $v_0 \leftarrow$ jaki wierzchołek z V
3. $V' \leftarrow \{v_0\}$
4. while $V' \neq V$

$E' \leftarrow E' \cup \{e\}$, gdzie e to najlżejsza krawędź incydentna z jednym wierzchołkiem z V' i jednym z $V \setminus V'$

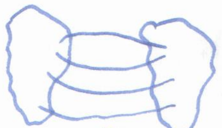
$V' \leftarrow V' \cup \{v_x\}$, v_x - koniec e

Jak dowieść poprawności algorytmu Kruskala?

1. Zauważmy, że wyprodukowane E' tworzy drzewo spinające.

D-d AA

E' składa się z ≥ 2 spójnych składowych

A  B

istnieją krawędzie między A i B, bo G jest spójny

No to weźmy najmniejszą z nich ($c(e)$ - minim.). Ale Kruskal musiał ją rozważać. Skoro teraz nie ist. cykl, to wtedy też ↯

2. Niech $e_1' \leq e_2' \leq \ldots \leq e_{n-1}'$ ← rozw. Kruskala

AA

Załóżmy, że ist OPT < Kruskala. Weźmy taki OPT, który $o_1, o_2, \ldots, o_{n-1}$ taki, który ma najdłuższy prefiks wspólny z Kruskalem.

Załóżmy, że

$e_1' = o_1$

$\vdots$ } w sensie krawędzi nie wag oraz $e_i' \neq o_i$

$e_{i-1}' = o_{i-1}$

1. $c(e_i') \leq c(o_i)$

$e_1 \ldots e_{i-1} \quad e_i$

$\;\; = \qquad\;\; = \qquad \wedge$, dorzućmy e_i do opta

$o_1 \ldots o_{i-1} \quad o_i$

$OPT \cup \{e_i\}$

W tym cyklu jest jakaś o_j (dla $j > i$)

$c(o_j) \geq c(e_i)$

$OPT \cup \{e_i\} \setminus \{o_j\}$ jest drzewem. Jeśli $c(e_i) < c(o_j)$, to drzewo ma koszt < OPT ↯, a jeśli $c(e_i) = c(o_j)$, to drzewo ma dłuższy prefiks z Kruskalem ↯

2. $c(e_i) > c(o_i)$

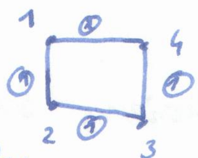
Wówczas krawędź o_i byłaby rozważona przez Kruskala wcześniej niż e_i. Ponieważ nie tworzy ona cyklu z $e_1, \dots, e_{i-1}$ byłaby wzięta do rozwiązania.

$\boxed{\text{Algorytm Borůvki}}$

1° $E' \leftarrow \emptyset$

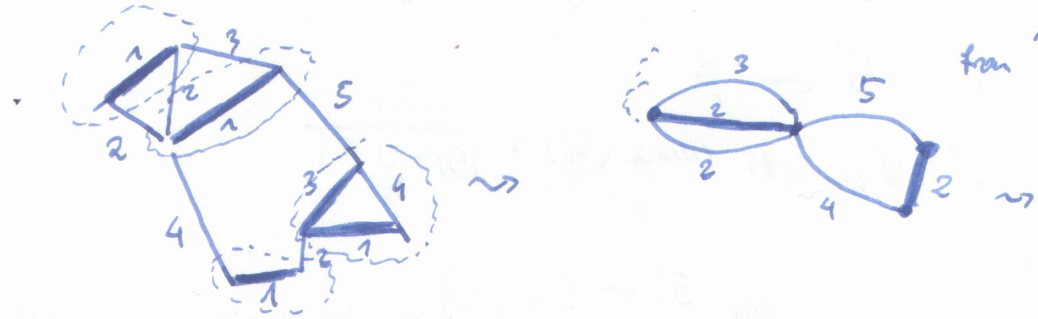
2° $\forall v \in V$ znajdujemy najtańszą krawędź incydentną z v

3° $E' \leftarrow E' \cup \{e_v, v \in V\}$.

Teoretycznie może powstać cykl 

Wystarczy, że dodamy jeszcze drugie kryterium, np. $\langle 1, 1, 4 \rangle$

from waga to



4° Tworzymy nowy graf $G_1 = (V_1, E_1)$

$V_1 = \{x : x \text{ jest spójną składową w } E'\}$

$\forall a,b \in V$ $\{a,b\} \in E_1$, iff istnieje krawędź jednym z wierzchołkiem z a i drugim wierzchołkiem z b

Powtarzaj tak długo jak $|V_1| > 1$

Problem (Pokrycie zbioru)

Dane: $S=\{S_1,S_2,\dots,S_n\}$ — podzbiory n-elementowego uniwersum

f-cja kosztu: $c: S_i \to R_i$

Żądane: znaleźć podrodzinę $S' \subseteq S$ t. że

$\bigcup_{X\in S'} X = \mathcal{U}$ oraz Koszt(S') zdefiniowany jako: $\sum_{X\in S'} c(X)$ jest minimalny

Fakt: TEN PROBLEM JEST NP-ZUPEŁNY

Pokażemy strategię zachłanną, która daje strategię o koszcie nie większym od log(n)OPT

1° $S' \leftarrow \emptyset$

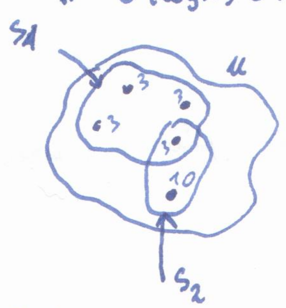
2° $\forall_{zb.\ S_i \notin S'}$ $\text{cena}(S_i) = \dfrac{c(S_i)}{|S_i \setminus \bigcup_{T\in S'} T|}$

$S' \leftarrow S \cup \{S_i\}$ ← takie że cena(S_i) jest minimalna i $S_i \notin S'$

Lemat

Rozwiązanie znalezione przez ten algorytm zachłanny ma koszt $\leq O(\log n)$ OPT

Przykład



$c(S_1)=12$, $c(S_2)=10$

cena$(S_1)=3$, cena$(S_2)=5$

Najpierw bierzemy S_1, potem S_2, ale za S_2 płacimy 10 mimo że pokrywamy de facto jeden element

D-d

Niech $e_1, e_2, \dots, e_n$ - elementy $\mathcal{U}$ w kolejności ich pokrywania. Zastanówmy się jaka jest cena pokrycia e_i. Oszacujmy ją. Jak oszacujemy każdy, to oszacowanie rozwiązania to jest suma po tych oszacowaniach.

Zauważmy, że tzn. że zawsze bierzemy spójny podciąg tego ciągu.

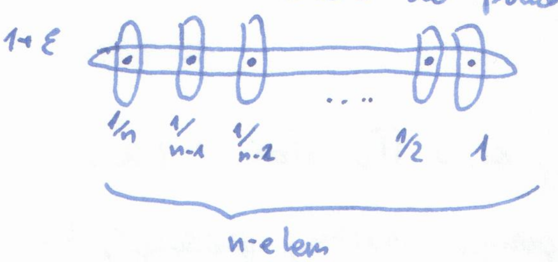
Gdy e_i był pokrywany elementów niepokrytych było $n-(i-1)=n-i+1$.

Stąd $\text{cena}(e_i) \leq \frac{OPT}{n-i+1}$. Tak więc koszt rozwiązania $\text{Koszt}(S') =$

$$= \sum_{i=1}^{n} \text{cena}(e_i) \leq \sum_{i=1}^{n} OPT \frac{1}{n-i+1} = OPT\left(1+\frac{1}{2}+\frac{1}{3}+\ldots+\frac{1}{n}\right) = OPT \cdot O(\log n)$$

Czy nie można lepiej? Tzn. czy nie można otrzymać lepszego wsp. aproksymacyjnego? A może nasz algorytm jest lepszy niż się nam wydaje?

To pierwsze pytanie jest dość trudne, natomiast to ostatnie jest nie bardzo trudne do pokazania, że nie jest lepszy niż szacujemy.



Weźmiemy $\frac{1}{n}$, potem $\frac{1}{n-1}$, itd. Zatem zachłannie: $\frac{1}{n}+\frac{1}{n-1}+\ldots+1$. No a opt. jest $1+\varepsilon$

$$\boxed{\text{Problem (szeregowanie zadań z terminami)}}$$

<u>Dane</u>: $d_1, \ldots, d_n \in \mathbb{N}$ - deadlines

$g_1, \ldots, g_n \in \mathbb{R}_+$ - zyski

(interp: mamy n zadań o jednostkowym czasie wykonania)

potem nie mamy już zysku dla $d_i = 2$

Za wykonanie zadania w terminie otrzymujemy zysk

<u>Zadanie</u>: Znaleźć podzbiór zadań maksymalizujący zysk

<u>Strategia zachłanna</u>:

$S \leftarrow \emptyset$

Niech zadania będą szeregowane nierosnąco wg zysków

Niech z_i to kolejne zadanie

Jeśli $S \cup \{z_i\}$ jest wykonywalny to

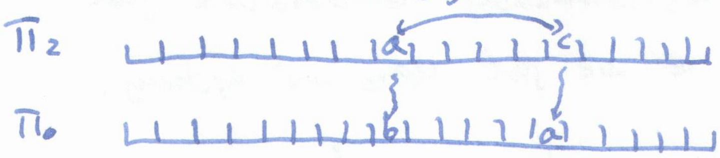
$S \leftarrow S \cup \{z_i\}$

Zbiór zadań jest wykonywalny, jeśli jego elementy można ustawić w ciąg t. że i-ty element ma deadline $\geq i$

<u>D-d</u> że nasz alg. daje opta

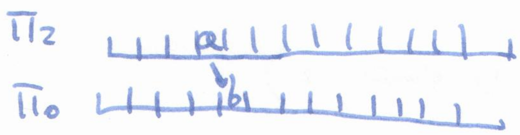
Niech Z - zbiór zadań znaleziony przez alg zachł., a O - zbiór zadań znaleziony przez alg optymalny. Niech Π_Z - ciąg wykonywania dla Z, i podobnie Π_O.

1. Bso możemy założyć, że każde zadanie z $Z \cap O$ jest na takiej samej pozycji w Π_Z i w Π_O.



Niech $a \in Z \cap O$. Powiedzmy, że w miejscu a w Π_O stoi $b \neq a$. Ale możemy zamienić a i c, bo do tej pozycji możemy przesunąć, bo w Π_O jest OK. Zadania są koło siebie. Dziury nie mają sensu, bo możemy zsunąć i zad. i tak będą wykonalne

2. Pokażemy, że na pozostałych pozycjach są zadania o tym samym zysku.



(i) $g_a > g_b$:

niemożliwe, bo $O \setminus \{b\} \cup \{a\}$ jest zbiorem wykonywalnym i

zysk $(O \setminus \{b\} \cup \{a\}) >$ zysk (O)

(ii) $g_a < g_b$

Wówczas algorytm zachłanny rozpatrywałby z przed a i umieściłby b przed a i umieściłby je w rozwiązaniu.

$Z \setminus \{a\} \cup \{b\}$ jest wykonywalny oraz $Z' \subseteq Z$ też.