Dolne granice

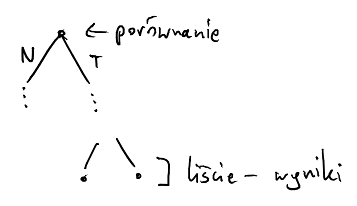
- modele:
  - maszyna RAM
  - maszyna Turinga
  - rachunek lambda

  } trudne
  - ⋮
  - model drzew decyzyjnych ← tym się zajmiemy

Model drzew decyzyjnych

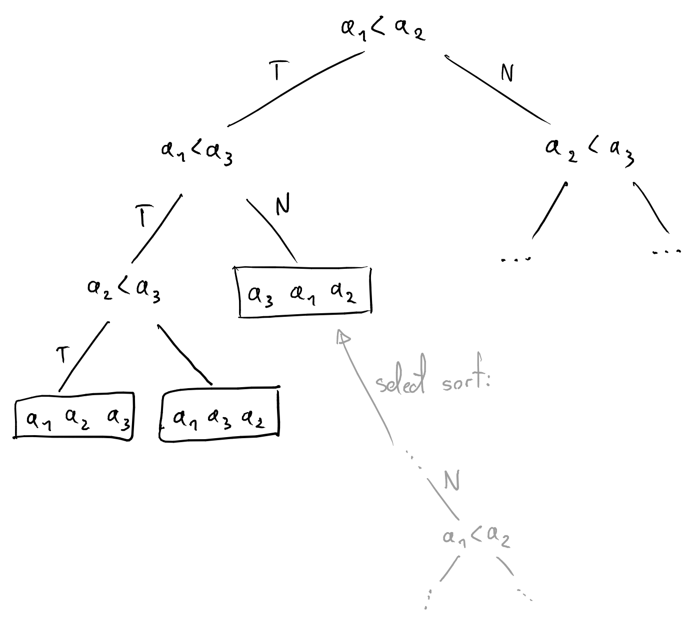
Dane mogą być użyte jedynie w porównaniach

Drzewo decyzyjne



Przykład

uporządkować $a_1\ a_2\ a_3$



jeden algorytm ~ rodzina drzew decyzyjnych (← nieskończona)
(drzewo dla każdego rozmiaru danych)

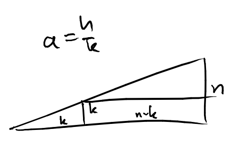
Fakt

Niech A - algorytm sortujący
$\{D_n\}_n^\infty$ - odpowiadająca rodzina drzew decyzyjnych

Wówczas każde drzewo D_n posiada co najmniej $n!$ liści.
(a zatem na wysokość $\Omega(n \log n)$)

$\log(0!) = 0$
$\log(1!) = 0$
$\log(2!) = 1$
$\log(3!) =$

$\log(n!) = \Omega(n \log n)$
$\log(n^n) = n \log n$
$\log((\frac{n}{2})^{\frac{n}{2}}) = \frac{n}{2}\log(\frac{n}{2}) = \frac{1}{2}n(\log n - \log 2)$
$\log\left(\left(\frac{n}{k}\right)^{\frac{k-n}{k}}\right)$ $\quad a = \frac{n}{k}$



Wniosek

Każdy algorytm sortujący za pomocą porównań musi wykonać co najmniej $c \cdot n \log n$ porównań sortując ciąg n-elementowy.

Jeśli ograniczymy się do permutacji $\sigma_i \in S_n$, to średnia złożoność może być wyrażona jako

$\sum_{\sigma \in S_n} \sigma \cdot d_\sigma$ ← gł. liścia dla σ

?... $\Omega(n \log n)$

Problem: różność elementów

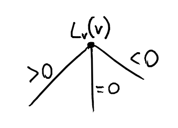
Dane: $x_1, \ldots x_n \in R$

Zadanie: sprawdzić czy $\forall_{i,j}.\ x_i \neq x_j$

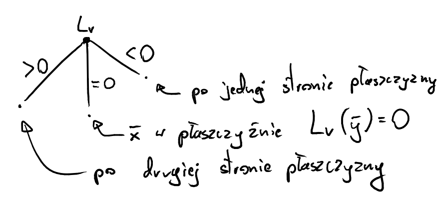
→ techniki powyżej nie przynoszą nic ciekawego, bo mamy tylko 2 wyniki

Liniowe drzewa decyzyjne

- drzewo ternarne (trynarne?)
- w wierzchołku v: kombinacja liniowa $L_v(\bar{x})$ [zamiast porównania]



- uogólnienie drzew decyzyjnych: $a_i < a_j \to a_i - a_j < 0$



~ ciachamy przestrzeń płaszczyznami

Def

$F(v) = \{\bar{x} \in R \mid \bar{x} \to v\}$

Obserwacja

$\forall v.\ F(v)$ jest zbiorem wypukłym.
(a zatem spójnym)

Tw? → do problemu wyżej?

Niech $P_i \in R^n$ $i \in [1..n!]$ t.że ich współrzędne tworzą różne permutacje $[1..n]$

Niech v_i będzie liściem T_n : $P_i \in F(v_i)$ $[P_i \to v_i]$

Wówczas $\forall i \neq j.\ v_i \neq v_j$

D-d

Niech $i \neq j$

P_i _ _ _ k ...
P_j _ k _ _ ...

k - najmniejsza liczba, która w P_i oraz P_j występuje na różnych pozycjach, odpowiednio k_i oraz k_j.

P_i _ a _ k ...
P_j _ k _ b ...
$\quad\ \ \uparrow k_j \quad \uparrow k_i$

$\boxed{a, b > k}$

Niech $f : \mathbb{R}^n \to \mathbb{R}$

$f(\langle x_i \rangle) = x_{k_i} - x_{k_j}$

$f(P_i) < 0$

$f(P_j) > 0$

Nie wprost: $\exists v$- liść. $P_i, P_j \in F(v)$ $\Rightarrow F(V) \subseteq \{\bar{x} \in \mathbb{R}^n \mid \forall i,j.\ \bar{x}[i] \neq \bar{x}[j]\}$

$\exists Q \in [P_i, P_j].\ f(Q) = 0$ ← f - ciągła

$Q \in F(v)$ [wypukłość]

$Q[k_i] = Q[k_j]$ $\lightning$ $Q \in F(v)$

$[P_i, P_j \to \text{TAK},\ Q \to \text{NIE}]$

Wniosek: algorytm dla różności elementów jest $\Omega(n \log n)$

---

otoczka wypukła