ASD, wykład 07, 2022.4.7

Dolne ograniczenie złożoności problemu sortowania

Nie istnieje algorytm sortowania oparty o porównania elementów, który wykonywałby dla każdych danych wejściowych $o(n \log_2(n))$ porównań.

Inaczej, dla każdego algorytmu sortowania opartego o porównania istnieje $\varepsilon > 0$ t. ż. złożoność pesymistyczna tego algorytmu $\geq \varepsilon \cdot n \log_2(n)$. (~~licz~~ złożoność liczona ilością porównań).

Co to znaczy algorytm sortujący oparty o porównania?

Def. A jest alg. ~~sp~~ sort. opartym o porównania jeśli dostęp do tablicy realizuje się wyłącznie przez instrukcje typu $t[i] \leq t[j]$. (i przez swap(t, i, j))

Algorytm może tylko sprawdzić względny porządek dwóch elementów. (i zamieniać je miejscami.)

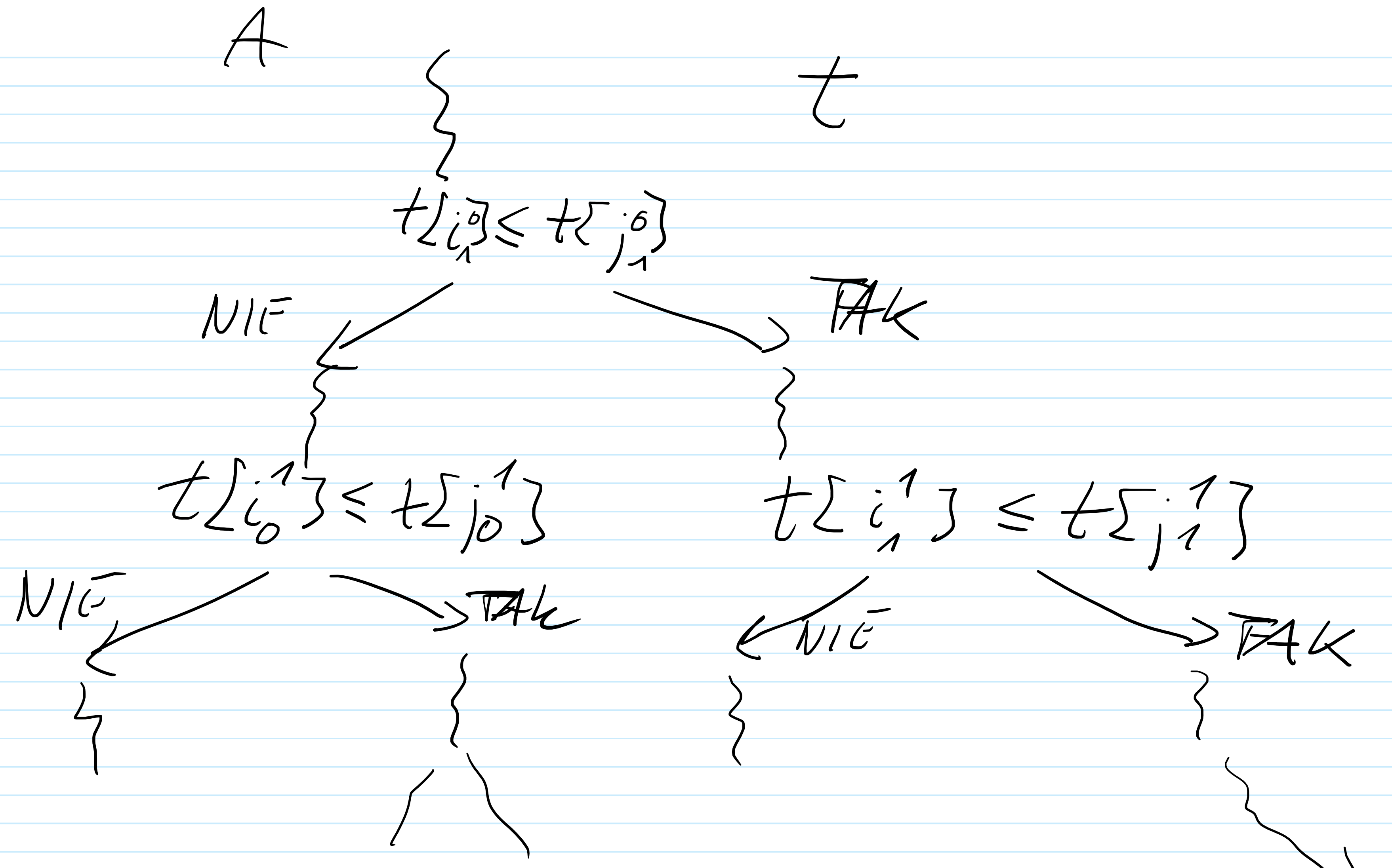
Wszystkie znane nam algorytmy sortujące mają tę własność.

Tw

Niech A - alg. sortujący oparty o porównania.
Istnieje stała $\varepsilon > 0$ (niezależna od A)
pesymistyczna złożoność A $\geq \varepsilon \cdot n \cdot \log_2(n)$.

Dow

Zmiana działania A spowodowana wejściem
następuje tylko w wyniku odpowiedzi na
pytanie postaci $t[i] \leq t[j]$ i mamy, wtedy
rozgałęzienie obliczenia na dwie ścieżki
zależne od odpowiedzi TAK lub NIE.



posortowane

liść w drzewie obliczenia.

(1) Wszystkie możliwe obliczenia
są reprezentowane w powyższym drzewie.

(2) Liście tego drzewa to wyjście algorytmu -
- posortowane tablice.

Załóżmy, że sortujemy tablicę elementów
1, 2, 3, ..., n. (ułożonych dowolnie).

Ile musimy mieć liści w drzewie
wszystkich możliwych obliczeń algorytmu A?

Każdy liść (posortowana tablica) to pewna
permutacja wejścia) i co więcej jeśli sortujemy
elementy 1, 2, ..., n to każda permutacja jest możliwym wyjściem.

Bo dla każdego wymieszania elementów 1, 2, ..., n jest tylko jedna permutacja która te ~~elem~~ wymieszane elementy porządkuje.

Wniosek

Nasze drzewo obliczeń ma przynajmniej $n!$ liści. (bo jest tyle wymieszań).

Mamy drzewo binarne które ma $\geq n!$ liści.

Jaka jest jego głębokość??

U ~~tes~~ nas głębokości drzewa

to ilość pytań zadanych na

najdłuższej ścieżce od korzenia do liścia.

Czyli jest to pesymistyczna złożoność

~~od~~ algorytmu A (złożoność liczona

ilością porównań).

$n! \approx \sqrt{2\pi n} \left(\frac{n}{e}\right)^n$ (Stirling)

$n! \geq (n)^{n/2}$, dla $n \geq 10$. (u mnie $n \geq 4$)

Mamy przynajmniej

$$n! \geq (n)^{\frac{n}{2}} \text{ liści.}$$

Jeśli drzewo ma głębokość k to ma

$\leq 2^k$ - liści

głębokość k drzewa musi spełniać

$$2^k \geq n! \geq (n)^{\frac{n}{2}}$$

$$\cancel{\log}\ \underbrace{k \geq \frac{n}{2} \cdot \log_2(n)}$$

czyli najdłuższa, (pesymistyczna) ścieżka

obliczenia zadaje

$\geq \frac{1}{2} n \log_2(n)$ zapytań $\square$