Lista 3
Zadanie 7

Algorytmy i Struktury Danych

Treść
Wykaż, że nie istnieje algorytm sortujący, który działa w czasie liniowym dla co najmniej połowy $n!$ możliwych danych wejściowych długości n. Czy odpowiedź ulegnie zmianie, jeśli zapytamy o ułamek $\frac{1}{n}$ i $\frac{1}{2^n}$ wszystkich permutacji?

Rozwiązanie
Do rozwiązania tego zadania przyda nam się znajomość ograniczenia $n!$:

$$\left(\frac{n}{e}\right)^n e \leq n! \leq e\left(\frac{n+1}{e}\right)^{n+1} e. \tag{1}$$

Na początku rozważymy drzewo decyzyjne o wysokości h, które odpowiada algorytmowi sortującemu n elementów. Liczba permutacji danych wejściowych długości n wynosi $n!$, zatem rozważane przez nas drzewo decyzyjne musi mieć co najmniej $n!$ liści. Jednocześnie drzewo o wyskości h nie może mieć więcej niż 2^h liści.
Dla co najmniej połowy z $n!$ możliwych permutacji danych wejściowych długości n mamy zatem

$$\frac{n!}{2} \leq 2^h.$$

Po nałożeniu $\log_2$ na obie strony nierówności dostajemy

$$\log_2(n!) - \log_2(2) \leq h\log_2(2).$$

Wiedząc, że $\log_2(2) = 1$ otrzymujemy natępujące wyrażenie

$$h \geq \log_2(n!) - 1.$$

Skorzystamy teraz z lewej strony nierówności (1) do ograniczenia wartości $n!$

$$h \geq n\log_2(n) - n\log_2(e) + \log_2(e) - 1 = \Omega(n\log_2(n)),$$

czyli $h = \Omega(n\log_2(n))$. Dla ułamka $\frac{1}{n}$ otrzymujemy

$$\begin{aligned}\frac{n!}{n} &\leq 2^h\\ h &\geq \log_2(n!) - \log_2(n)\\ h &\geq n\log_2(n) - n\log_2(e) + \log_2(e) - \log_2(n) = \Omega(n\log_2(n)).\end{aligned}$$

Natomiast dla $\frac{1}{2^n}$ również otrzymujemy:

$$\begin{aligned}\frac{n!}{2^n} &\leq 2^h\\ h &\geq \log_2(n!) - n\log_2(2)\\ h &\geq n\log_2(n) - n\log_2(e) + \log_2(e) - n = \Omega(n\log_2(n)).\end{aligned}$$

W żadnym przypadku nie jesteśmy w stanie otrzymać liniowości.