Algorytmy i struktury danych

Zadanie 6, lista 1

Dawid Żywczak

6 kwietnia 2020

Na wejściu dostajemy ciąg $a_1, a_2, ..., a_n$ taki, że dla każdego i $a_i <= a_{i+1}$. Jedyna operacja jaką możemy na tym ciągu wykonywać, to usuwanie elementów a_i oraz a_j, gdy $2a_i \leq a_j$. Mamy skonstruować algorytm, obliczający ile conajwyżej par możemy usunąć. Problem ten rozwiążemy algorytmem zachłannym.

$i \leftarrow 1$
$j \leftarrow \lfloor \frac{n}{2} \rfloor + 1$
$k = 0$
while $j \leq n$ **do**
  if $2A[i] \leq A[j]$ **then**
    $delete\_pair(a_i, a_j)$
    $k \leftarrow k + 1$
    $i \leftarrow i + 1$
  end if
  $j \leftarrow j + 1$
end while
return k

Co my tak naprawdę robimy? Zachłannie szukamy pary dla elementów prawej części tablicy. Jeżeli się udaje, usuwamy daną parę i zwiększami nasz licznik usuniętych par o 1. Dlaczego to działa?
Załóżmy, że mamy rozwiązanie optymalne, nazwijmy je O, które usuwa k par. Chcę pokazać, że jesteśmy w stanie przekształcić rozwiązanie O na rozwiązanie produkowane przez powyższy algorytm. Jak to zrobić?

Lemat 1. *Niech L i R będą zbiorami, takimi że dla dowolnej usuwanej pary* (a_i, a_j), $a_i \in L$, $a_j \in R$, *zbiór L posiada k elementów, oraz indeksy elemntów w zbiorze R są* $\leq \lfloor \frac{n}{2} \rfloor$. *Chcemy też by zbiory L i R spełniały założenia zadania, tzn każdy element zbioru L jest mniejszy od najmniejszego elementu ze zbioru R.*

Dowód. Jak to pokazać? Rozważmy przypadki!
1. Weźmy dowolne pary punktów, które możemy usunąć (a_i, a_{i1}) oraz $(a_j, a_{j1}$, takie że $a_{i1} \leq a_j$. Wtedy pary (a_i, a_j) oraz (a_{i1}, a_{j1}) również możemy usunąć, a z tego wynika,

że możemy je ułożyć w wyżej opisany sposób.
2. Weźmy dowolny punkt a_i o indeksie mniejszym od k. Powinnien należeć do zbioru L, ale załóżmy, że tak nie jest. Weźmy najmniejszy punkt $a_j \in L$ o większym indeksie od i. Wtedy można punkt sparowany z nim sparować z naszym a_i, a punkt a_j sparować z parą kolejnego punktu i postępować w ten sposób tak długo, aż każdy punkt będzie miał parę.
3. A co jeśli w zbiorze R znajdują się jakieś elementy o indeksie mniejszym od m? Możemy je zamienić na elementy o większych indeksach $\geq m$ korzystając z własności naszego ciągu oraz faktu, że $k \leq m$. □

Teraz wystarczy pokazać, że dla każdego elemntu wyżej zdefiniowanego zbioru L, nasz algorytm znajudje parę.

Dowód. Baza: Element a_1 zostanie sparowany z elementem ze zbioru R o najmniejszym indeksie, zatem znajdzie parę z najmniejszym elementem R. Krok: Weźmy dowolne $i \leq k$ i załóżmy, że znaleźliśmy pary dla wszystkich elemntów $a_1, a_2, ..., a_{i-1}$. Znalezione pary, to kolejne najmniejsze elementy ze zbioru R, zatem w szczególności są one $\geq a_m$. Wtedy punkt a_i znajdzie sobie parę z kojenym najmniejszym punktem, po parze punktu a_i, który spełni wymaganą nierówność. Taki punkt na pewno istnieje, bo możemy brać też elementy ze zbioru R. □

Jaką mamy złożoność? Maksymalnie n razy sprawdzimy nasz warunek przechodząc całą tablicę, więc jest to O(n).