Ćwiczenia 5

Zasada 0–1 dla sieci sortujących ciągi bitoniczne

Tw. Sieć sortująca sortuje dowolne ciągi bitoniczne wtw, gdy sortuje ciągi binarne ciągi bitoniczne.

Niech S sieć sortująca o n wejściach, która poprawnie wszystkie binarne ciągi bitoniczne. Niech $a_1, a_2, \ldots, a_n$ bitoniczny ciąg (liczb naturalnych), dla którego sieć daje błędny wynik, np. $a'_1, a'_2, \ldots, a'_n$ ($\{a'_i\}$ — permutacja $\{a_i\}$), t.że istnieje $1 \leq x < y \leq n$, $a'_x > a'_y$.

Niech funkcja f będzie zdefiniowana w następujący sposób:

- $f(z) = 0$, gdy $z < a'_x$,
- $f(z) = 1$, wpp.

Przy takiej definicji f ciąg $f(a_1), f(a_2), \ldots, f(a_n)$ jest binarnym ciągiem bitonicznym. Z lematu o funkcji niemalejącej wynika, że sieć S dla powyższego ciągu zwróci wynik $f(a'_1), f(a'_2), \ldots, f(a'_n)$. Tak jak poprzednio ten wynik nie jest poprawny dla $f(a'_x) = 1 > f(a'_y) = 0$ i $x < y$. Więc sprzeczność — S nie sortuje wszystkich binarnych ciągów bitonicznych (w szczególności nie sortuje $f(a_1), f(a_2), \ldots, f(a_n)$).

Udowodnij, że sieć komparatorów poprawnie sortuje ciąg $n, n{-}1, \ldots, 1$ wtw., gdy poprawnie sortuje ciągi $1^i\,0^{n-i}$ $(0 \leq i \leq n)$.

Dowód przez sprzeczność:

- niech sieć daje błędny wynik dla ciągu $n, n{-}1, \ldots, 1$, ale sortuje poprawnie wszystkie ciągi $1^i 0^{n-1}$. Niech wynikiem dla ciągu $n, n-1, \ldots, 1$ będzie ciąg $a_1, a_2, \ldots, a_n$, oraz istnieje $i < j$, t.że $a_i > a_j$. Można zastosować rozumowanie analogiczne do poprzedniego punktu, niech $f(k) = 0$ wtw $k < a_i$, wpp $f(k) = 1$. Po zastosowaniu funkcji f do ciągu $n, n-1, \ldots 1$, powstaje ciąg $1^{(n+1-a_i)} 0^{a_i - 1}$ (który jest poprawnym ciągiem typu $1^i 0^{n-i}$). Jednak po zastosowaniu funkcji f do wyniku otrzymywany jest pewien ciąg binarny $b_1, b_2, \ldots, b_n$ ($b_x = f(a_x)$), w którym $b_i = f(a_i) = 1$, $b_j = f(a_j) = 0$ (ponieważ $a_i > a_j$), stąd znów $i < j$ i $b_i > b_j$. Tak więc sieć nie sortuje również poprawnie ciągów typu $1^i 0^{n-i}$.
- niech sieć sortuje poprawnie ciąg $n, n-1, \ldots, 1$ ale istnieje i, t.że dla ciągu $1^i 0^{n-i}$ generowany jest zły wynik. Niech $f(x) = 1$ wtw, gdy $x \geq i$, $f(x) = 0$ wpp. Po zastosowaniu funkcji f do $n, n-1, \ldots, 1$ otrzymywany jest właśnie ciąg $1^i 0^{n-i}$, a po zastosowaniu jej do wyniku sortowania pierwotnego ciągu $0^{n-i} 1^i$. Stąd wynika, że jednak sieć poprawnie sortuje ciąg $1^i 0^{n-i}$ — sprzeczność.

Udowodnij, że jeśli sieć transpozycjna sortuje, to ma $\Omega(n^2)$ procesorów.

Niech $a_1, a_2, \ldots, a_k$ dane wejściowe, a k — liczba inwersji w ciągu $\{a_i\}$. Kluczowa jest obserwacja, że liczba inwersji po przejściu przez jeden komparator, nigdy nie ulegnie zwiększeniu, a może zostać zmniejszona co najwyżej o 1. Niech komparator łączy a_i i a_{i+1}:

- jeśli $a_i \leq a_{i+1}$ to na wyjściu komparatora ciąg nie ulega żadnym zmianom a tym samym liczba inwersji się nie zmienia,
- jeśli $a_i > a_{i+1}$ to elementy są zamieniane, tym samym znika jedna inwersja (a_i, a_{i+1}).

Ponieważ ciąg wejściowy może mieć $O(n^2)$ inwersji (np. ciąg $n, n-1, \ldots 1$ ma ich dokładnie $n(n-1)/2$). Stąd komparatorów również musi być (co najmniej) tyle.

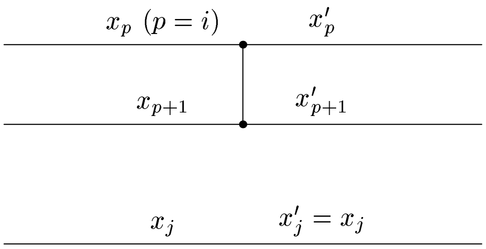
Udowodnij, że sieć transpozycjna sortuje wtw, gdy sortuje ciąg $n, n-1, \ldots 1$.

Dowód powyższej własności wynika z następującego faktu:
Fakt: Niech S transpozycjna sieć sortująca, $\{x_k\}$–dane wejściowe, $\{x'_k\}$–wynik, jeśli dla pewnych i,j, $(i < j)$ $x'_i > x'_j$, to w wyniku działania sieci — $\{y'_k\}$ dla danych $\{n, n-1, \ldots, 1\}$ również występuje niezgodność, tzn. $y'_i > y'_j$.

Dowód: Można posłużyć się indukcją po liczbie komparatorów.
Dla $n = 0$: własność jest spełniona — dla dowolnych $i < j$: $y'_i > y'_j$.
Dla $n > 0$: niech ostatni komparator łączy linie p i $p+1$, trzeba pokazać, że jeśli dla $i < j$ $x'_i > x'_j$, to $y'_i > y'_j$ ($\{x_k\}$–dane na liniach przed komparatorem, $\{x'_k\}$—dane po uwzględneniu ostatniego komparatora, $\{y_k\}$, $\{y'_k\}$—analogicznie dla symulacji sortowania ciągu $\{n, n-1, \ldots, 1\}$).

Należy rozważyć następujące przypadki:

- $i = p$, $j > p + 1$ (j nie może być równe p, gdyż $x'_p < x'_{p+1}$),



Rys. 1. Fragment sieci dla pierwszego przypadku

$x'_i = x'_p > x'_j = x_j \;\rightarrow\; x'_p, x'_{p+1} > x_j \;\rightarrow\; x_p, x_{p+1} > x_j \;\rightarrow\; y_p, y_{p+1} > y_j$

Z założenia indukcyjnego:

$$x_p, x_{p+1} > x_j \quad\rightarrow\quad y_p, y_{p+1} > y_j \quad\rightarrow\quad y'_p, y'_{p+1} > y_j = y'_j$$

- $i = p+1$, $j > p+1$,

$$x'_j = x'_{p+1} > x'_j = x_j \quad \rightarrow \quad \max(x_p, x_{p+1}) > x_j$$

z założenia indukcyjnego:

$$\max(x_p, x_{p+1}) > x_j \;\rightarrow\; \max(y_p, y_{p+1}) > y_j \;\rightarrow\; y'_{p+1} > y_j = y'_j$$

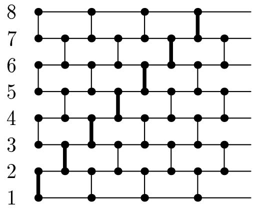
- analogiczne rozumowanie można przeprowadzić dla ($i < p$, $j = p$) i ($i < p$, $j = p+1$),
- jeśli $i, j \notin \{p, p+1\}$ teza wynika wprost z założenia indukcyjnego.

Wracając do pierwotnego problemu. Dowód w jedną stronę jest prosty, jeśli sieć sortuje, to tym oczywiście sortuje również ciąg $n, n-1, \ldots, 1$. W drugą stronę, można udowodnić tezę doprowadzając do sprzeczności. Niech sieć sortuje poprawnie ciąg $\{y_k\} = \{n, n-1, \ldots, 1\}$, ale dla pewnego ciągu $a_1, a_2, \ldots, a_n$ daje nieprawidłową odpowiedź, np. $i < j$, $a'_i > a'_j$. Jednak z właśnie udowodnionego faktu wynika, że w takim wypadku, również $y'_i > y'_j$, więc ciąg $n, n-1, \ldots 1$, również nie został posortowany — sprzeczność.

Udowodnij, że sieć odd–even sortuje.

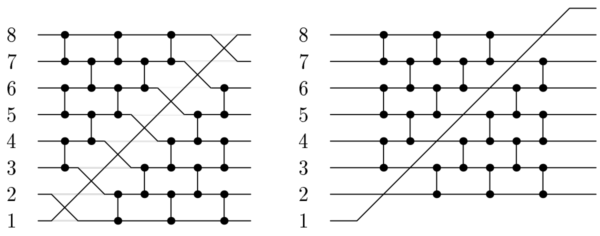
Z poprzedniego zadania wynika, że wystarczy pokazać, że sieć sortuje ciągi $n, n-1, \ldots, 1$. Można ten fakt pokazać przez indukcję:

- dla $n = 1$ — sieć oczywiście sortuje,
- pozostaje pokazać, że dla $n > 1$ sieć sortuje (przy założeniu, że wszystkie mniejsze sieci sortują).



Rys. 2. Sieć odd–even dla $n = 8$ i danych $8, 7, 6, 5, 4, 3, 2, 1$, pogrubioną linią zaznaczono komparatory w których porównywana jest wartość 1

Należy zauważyć, że wyniki porównań w których bierze udział wartość 1, są z góry ustalone — jedynka zawsze przechodzi poziom wyżej, a porównywana wartość — poziom niżej. Stąd można uprościć rysunek do postaci:



Rys. 3. Uproszczona sieć odd–even

Po uproszczeniu i wyprostowaniu wszystkich linii pozostaje sieć odd–even o rozmiarze $n-1$, która ma za zadanie posortować ciąg $n, n-1, \ldots, 1$. Ponieważ z założenia indukcyjnego mniejsze sieci sortują, a element 1 wraca na swoje miejsce w uporządkowanym ciągu, stąd i większa sieć sortuje.

Udowodnić, że każda sieć scalająca jedną liczbę z ciągiem posortowanym (o długości n) ma $\Omega(n)$ komparatorów i głębokość $\Omega(\log n)$

- niech sieć scalająca ma m komparatorów, i niech $m < n$, stąd pewna linia nie jest połączona (nawet pośrednio) z linią 0 (zawierająca element do scalenia). Zatem sieć nie jest w stanie "wsortować" liczby na tą linię, np. jeśli rozważana linią miałaby numer i, to dla danych $i, 1, 2, \ldots, i-1, i+1, i+2, \ldots, n$ sieć dawałaby nieprawidłową odpowiedź,
- aby udowodnić drugi fakt wystarczy zauważyć, że liczba pozycji na których może znaleźć się scalany element po przejściu przez i–ty komparator jest równa 2^i. Ponieważ scalany element może docelowo znajdować się na każdej z n pozycji, stąd i musi być rzędu $\Omega(\log n)$