Algorithm Correctness Proof.

Definition elimination : removal of given element.

Theorem 1

Κάθε array μπορεί να καταλήξει σε unique?

- Έστω ότι δεν μπορεί.

Έστω array μεγέθους n για το οποίο ισχύει $a_{i-1} > a_i > a_{i+1}$ και $a_1 < a_n$

$$a_i > a_{i+1} \Rightarrow a_i > a_n$$
$$a_n > a_1 \Rightarrow a_i > a_1$$
$$a_1 > a_{i-1} \Rightarrow a_i > a_{i-1} \text{ άτοπο } a_{i-1} < a_i \quad ∎$$

Leema 1

Ένας πίνακας n στοιχείων θα καταλήξει με 0 ή 1 στοιχεία μετά από k eliminations <u>αν δεν περιλαμβάνει διπλότυπα</u>

- Ο τελικός πίνακας μπορεί να έχει
  - 1 element αν $n > 0$
  - 0 elements αν $n = 0$

Proof.

για $n=0$ trivial

για $n \neq 0$:

λύση με $n>1$: $\begin{cases} \text{Έστω μετά από } k \text{ eliminations έχουμε έγκυρη βέλτιστη} \\ \text{μεγέθους } n>1 \text{. Χωρίς βλάβη επιλέγω } n=2. \\ \text{Άρα } a_1 = a_2 \\ \text{Άτοπο. Ο αρχικός πίνακας δεν είχε διπλότυπα.} \end{cases}$

λύση με $n=0$:

1) Έστω μετά από k eliminations έχουμε έγκυρη βέλτιστη μεγέθους $n=0$. Άτοπο καθώς με $k-1$ eliminations έχουμε επίσης βέλτιστη λύση με 1 στοιχείο

2) Επίσης για $n=1$ ισχύει $1 \leq i \leq 1$, και άρα $i+1$ δεν ορίζεται, οπότε elimination για $a_i < a_{i+1}$ δεν είναι εφικτό ∎

Theorem 2

$k = n-1$ για πίνακα μεγέθους n <u>χωρίς διπλότυπα.</u>

Proof

Από Θεώρημα 1 ισχύει ότι $\exists\, i \in [1, n-1]$ s.t $a_i \leq a_{i+1}$ για κάθε state του προβλήματος

Από Leema 1 ισχύει ότι για $n>0$ το τελικό state θα έχει μέγεθος 1 (0 είναι trivial)

Θέλουμε ακριβώς $n-1$ eliminations για να φτάσουμε σε πίνακα μεγέθους 1.

∎

Theorem 3

$k = n-1$ βέλτιστο για πίνακα μεγέθους n <u>χωρίς διπλότυπα.</u>

Proof

Έστω βέλτιστες λύσεις

για $k > n-1 \Rightarrow k = n$

Έστω μετά από k eliminations έχουμε έγκυρη βέλτιστη μεγέθους $n=0$. Άτοπο καθώς με $k-1$ eliminations έχουμε επίσης βέλτιστη λύση με 1 στοιχείο

για $k < n-1$.

Χωρίς βλάβη επιλέγω $k = n-2$

Άρα τελικός πίνακας με $n=2$ στοιχεία

- Είναι ίδια? Άτοπο καθώς αρχικός πίνακας χωρίς διπλότυπα
- Δεν είναι ίδια? : Άτοπο, η λύση ΔΕΝ είναι έγκυρη.

∎

Συνέπεια 1

Το Θεώρημα 1 ισχύει και για $a_{i-1} > a_i > a_{i+1}$ και $a_1 = a_n$

Απόδειξη trivial.

Theorem 4

$k = n-d$ για πίνακα μεγέθους n και d ο αριθμός των φορών που εμφανίζεται <u>το μοναδικό διπλότυπο νούμερο</u>

- Έστω πίνακας με $a_i = a_{i+j}$ και $a_p \neq a_q \ \forall p,q$ με $p \neq q$ και $p = i \wedge q = i+j$ ή αντίστροφα. (1). Επιλέγω $d=2$ χωρίς βλάβη της γενικότητας

$$A = [a_1 \cdots a_i \cdots a_{i+j} \cdots a_n]$$

- Από συνέπεια 1 του Θεωρήματος 1 και Θεώρημα 2 θέλουμε $e = \underbrace{i+j}-1-(i+1) = \cancel{i}+j-1-\cancel{i}-1 = j-2$ eliminations για να έχουμε

$$A' = [a_1 \cdots \overbrace{a_i, a_l, a_{i+j}}^{\text{ίδια}} \cdots a_n]$$

$\uparrow$ διαφορετικό.

με $a_i = a_{i+j}$ και $a_i \neq a_l$. (indexes είναι του A, και όχι του A' πίνακα).

(2) Αν $a_i < a_l$ επιλέγουμε το a_l για elimination

Αν $a_i > a_l$ τότε $a_l < a_{i+1}$ και κάνουμε το ίδιο

Αν $a_i = a_l$ αδύνατον από (1)

„Σε κάθε περίπτωση θέλουμε $e = \cancel{i}+j\cancel{-1} = j$ eliminations για να "κολλήσουμε" τα a_i, a_{i+j}.

- Μεταξύ $a_1 \ldots a_{i-1}$ ισχύει το Θεώρημα 3 με $e = i-1-1 = i-2$ για να έχουμε $[a_x, a_i, a_{i+j}, \ldots a_n]$.

- Μεταξύ $a_{i+j} \ldots a_n$ ισχύει το Θεώρημα 3 με $e = (n-i-j+1)-1$

- Καταλήγουμε με $e = \cancel{j}-2 \cancel{+i}-2+n\cancel{-i}\cancel{-j}\cancel{+1}\cancel{-1} = n-k$

$$[a_x, a_i, a_{i+j}, a_y]$$

αν $a_x < a_i$ και $a_{i+j} < a_y$ τότε με $k = n-d$ έχουμε βέλτιστη λύση

αν $a_x > a_i$ και $a_{i+j} < a_y$ τότε αφαιρούμε το a_y, και καθώς $a_{i+j} < a_x$ ($a_i = a_{i+j}$) αφαιρούμε το a_x

Παρόμοια με κάθε συνδυασμό, καταλήγουμε σε $k = n-d$.

Είναι εύκολο να δούμε ότι για $d > 2$ θα δουλεύει με τον ίδιο τρόπο. για $d=1$ θα δουλεύει από Θεώρημα 3.

∎

Theorem 5.

$k = n-d$ για πίνακα μεγέθους n και d ο αριθμός των φορών που εμφανίζεται το νούμερο με τις περισσότερες εμφανίσεις.

Έστω d_{max}, d με $d < d_{max}$.

Είναι trivial να δούμε ότι αν επιλέξουμε να "προσπατεύσουμε" διπλότυπα νούμερα με λιγότερες από τις max εμφανίσεις, θα κάνουμε $k = n-d$ eliminations, ενώ με max έχουμε $k_{min} = n - d_{max}$.

$$k > k_{min}.$$