Εφαρμοσμένη Συνδυαστική - Young Tableaux

Βασικοί ορισμοί

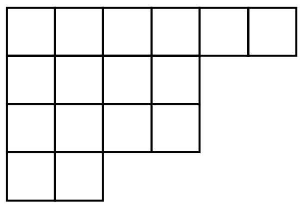
Διαμέριση ενός φυσικού αριθμού $n \in \mathbb{N}^*$ είναι μια m-άδα $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_m)$ θετικών ακεραίων, τέτοια ώστε

$$n = \lambda_1 + \lambda_2 + \cdots + \lambda_m, \qquad \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_m.$$

Το **διάγραμμα Young**, ή απλά διάγραμμα, που ορίζει η διαμέριση $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_m)$ είναι το σύνολο

$$[\lambda] := \{(i, j) : i \in [m], j \in [\lambda_i]\}$$

και αναπαριστάνεται σχηματικά ως ένα σύνολο από κελιά, διατεταγμένα σε αριστερά στοιχισμένες γραμμές, έτσι ώστε κάθε γραμμή να περιέχει το πολύ όσα κελιά περιέχει η προηγούμενή της. Για παράδειγμα, αν $n = 16$ και θεωρήσουμε τη διαμέριση $\lambda = (6, 4, 4, 2)$, τότε προκύπτει το διάγραμμα του Σχήματος 1.



Σχήμα 1: Το διάγραμμα της διαμέρισης $\lambda = (6, 4, 4, 2)$.

Ένα **Young tableau**, ή απλά tableau, προκύπτει αν γεμίσουμε τα κελιά ενός διαγράμματος με θετικούς ακεραίους, έτσι ώστε τα στοιχεία του να έχουν:

- *αύξουσα σειρά από αριστερά προς τα δεξιά σε κάθε γραμμή και*
- *γνησίως αύξουσα σειρά από πάνω προς τα κάτω σε κάθε στήλη.*

Το διάγραμμα που αντιστοιχεί σε ένα συγκεκριμένο tableau T, ονομάζεται **διάγραμμα** ή **σχήμα** του tableau και συμβολίζεται με $\mathrm{sh}(T)$. Πιο αυστηρά, ένα tableau T σχήματος $\mathrm{sh}(T) = [\lambda]$, όπου $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_m)$ διαμέριση του n, είναι μια απεικόνιση $T : [\lambda] \to \mathbb{N}^*$, τέτοια ώστε

$$j_1 < j_2 \Rightarrow T(i, j_1) \leq T(i, j_2) \qquad \text{και} \qquad i_1 < i_2 \Rightarrow T(i_1, j) < T(i_2, j)$$

για οποιαδήποτε κελιά του $[\lambda]$. Από εδώ και στο εξής, θα θεωρούμε ότι τα στοιχεία του tableau χρησιμοποιούν όλα τα στοιχεία ενός συνόλου $[k], k \leq n$, δηλαδή ότι $T([\lambda]) = [k] \subseteq [n]$. Αν η απεικόνιση T είναι ένα προς ένα και επί του $[n]$, τότε λέμε ότι το T είναι ένα **standard Young tableau** (SYT).

1	2	2	3	3	5
2	3	5	5		
4	4	6	6		
5	6				

1	3	7	12	13	15
2	5	10	14		
4	8	11	16		
6	9				

Σχήμα 2: Ένα tableau (αριστερά) και ένα standard tableau (δεξιά).

Το **συζυγές** διάγραμμα ενός διαγράμματος $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_r)$, συμβολίζεται με $\tilde{\lambda}$ και προκύπτει αν μετατρέψουμε τις γραμμές του λ σε στήλες και τις στήλες σε γραμμές, δηλαδή:

$$\tilde{\lambda} = (\mu_1, \mu_2, \ldots, \mu_s), \text{ με}$$

$$s = \lambda_1 \text{ και } \mu_i = |\{\lambda_j \geq i, j \in [r]\}|, \forall i \in [s].$$

Το **ανάστροφο** tableau ενός tableau T σχήματος λ, συμβολίζεται με $\mathbf{T}^\tau$ και προκύπτει αν μετατρέψουμε τις γραμμές του T σε στήλες και τις στήλες σε γραμμές, όπως προηγουμένως, έτσι ώστε το σχήμα του T^τ να είναι το $\tilde{\lambda}$ και για κάθε στοιχείο a_{ij} του T να ισχύει:

$$a_{ij} = b_{ji},$$

όπου b_{ji} είναι στοιχείο του T^τ.

Στο σύνολο των διαμερίσεων ορίζεται μια μερική διάταξη ως εξής: Αν $\mu = (\mu_1, \mu_2, \ldots, \mu_r)$ και $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_s)$ διαμερίσεις, τότε

$$\mu \leq \lambda \Leftrightarrow r \leq s \text{ και } \forall i \in [r], \mu_i \leq \lambda_i.$$

Στην περίπτωση αυτή, προφανώς είναι $[\mu] \subseteq [\lambda]$, οπότε λέμε ότι το διάγραμμα $[\mu]$ **περιέχεται** στο διάγραμμα $[\lambda]$. Το σύνολο των διαμερίσεων, εφοδιασμένο με αυτήν την μερική διάταξη, είναι το λεγόμενο δικτυωτό Young.

Ένα **skew διάγραμμα** προκύπτει αν από ένα διάγραμμα $[\lambda]$ αφαιρέσουμε ένα διάγραμμα $[\mu]$ με $\mu \leq \lambda$. Το διάγραμμα που προκύπτει συμβολίζεται $[\lambda] \setminus [\mu]$. Ένα **skew tableau** προκύπτει αν γεμίσουμε τα κελιά του skew διαγράμματος με τον τρόπο που περιγράψαμε νωρίτερα.

Σχήμα 3: Ένα skew διάγραμμα (αριστερά) και ένα skew tableau (δεξιά).

Ένα κελί το οποίο ανήκει στο $[\mu]$ (μαύρα κελιά), ονομάζεται **εσωτερική γωνία** του διαγράμματος $[\lambda] \setminus [\mu]$, όταν το κελί δεξιά από αυτό καθώς και το κελί κάτω από αυτό δεν ανήκουν στο $[\mu]$. Ένα κελί το οποίο ανήκει στο $[\lambda]$, ονομάζεται **εξωτερική γωνία** του διαγράμματος $[\lambda] \setminus [\mu]$, όταν το κελί δεξιά από αυτό καθώς και το κελί κάτω από αυτό δεν ανήκουν στο $[\lambda]$. Στο διάγραμμα του προηγούμενου σχήματος υπάρχουν 3 εσωτερικές γωνίες, τα κελιά $(1,4)$, $(2,3)$ και $(3,1)$ και 3 εξωτερικές γωνίες, τα κελιά $(1,5)$, $(3,4)$ και $(4,2)$.

Ο αλγόριθμος bumping

Ο αλγόριθμος bumping (ή row insertion) ορίζει μια διαδικασία εισαγωγής ενός νέου στοιχείου σε ένα υπάρχον tableau. Αν T είναι το υπάρχον tableau και x το νέο στοιχείο που θα εισαχθεί, τότε το νέο tableau που θα προκύψει συμβολίζεται ως $T \leftarrow x$ και θα έχει ένα επιπλέον κελί από το αρχικό tableau T.

Βήμα 1. Αν το x είναι μεγαλύτερο ή ίσο από κάθε στοιχείο της πρώτης γραμμής, τότε πρόσθεσε το x σε νέο κελί στο τέλος (δεξιά) της γραμμής και πήγαινε στο Βήμα 3.

Βήμα 2. Αν υπάρχει μεγαλύτερο στοιχείο από το x στην τρέχουσα γραμμή, τότε αντικατάστησε με το x το αριστερότερο στοιχείο της γραμμής που είναι μεγαλύτερο του x και επανάλαβε το Βήμα 1 για την επόμενη γραμμή και για το στοιχείο που αντικαταστάθηκε.

Βήμα 3. Τερματισμός.

Παράδειγμα: Έστω το tableau $T = \begin{array}{|c|c|c|c|c|}\hline 1&2&2&3&3\\ \hline 2&3&5 \\ \cline{1-3} 5&6 \\ \cline{1-2}\end{array}$ και έστω ότι θέλουμε να εισάγουμε το στοιχείο 2.

Παρακάτω φαίνονται τα βήματα του αλγορίθμου.

$$\begin{array}{|c|c|c|c|c|}\hline 1&2&2&3&3\\ \hline 2&3&5 \\ \cline{1-3} 5&6 \\ \cline{1-2}\end{array} \leftarrow 2 \qquad \begin{array}{|c|c|c|c|c|}\hline 1&2&2&2&3\\ \hline 2&3&5 \\ \cline{1-3} 5&6 \\ \cline{1-2}\end{array} \leftarrow 3 \qquad \begin{array}{|c|c|c|c|c|}\hline 1&2&2&2&3\\ \hline 2&3&3 \\ \cline{1-3} 5&6 \\ \cline{1-2}\end{array} \leftarrow 5$$

$$\begin{array}{|c|c|c|c|c|}\hline 1&2&2&2&3\\ \hline 2&3&3 \\ \cline{1-3} 5&5 \\ \cline{1-2} 6 \\ \cline{1-1}\end{array} = T\leftarrow 2$$

Είναι προφανές από τη διαδικασία ότι το αποτέλεσμα είναι πάντα ένα tableau γιατί δεν παραβιάζονται οι περιορισμοί του ορισμού των tableaux. Πράγματι, κατά την εισαγωγή ενός στοιχείου σε κάποια γραμμή, τα στοιχεία της γραμμής εξακολουθούν να έχουν αύξουσα σειρά. Επιπλέον, όταν ένα στοιχείο y εκτοπίζει από μια γραμμή ένα στοιχείο $z > y$, το στοιχείο x κάτω από το z (αν υπάρχει) είναι εξ ορισμού μεγαλύτερο του z και έτσι το z θα εκτοπίσει το x ή κάποιο αριστερότερο του x. Το στοιχείο πάνω από τη νέα θέση του z είναι μικρότερο ή ίσο του y άρα και μικρότερο του z. Επομένως και οι στήλες διατηρούν την γνησίως αύξουσα διάταξη.

Η αντίστροφη διαδικασία, δηλαδή αυτή της ανάκτησης του αρχικού tableau, είναι εφικτή εφόσον γνωρίζουμε το τελικό tableau και το νέο κελί που προστέθηκε στο διάγραμμα. Αρκεί να τρέξουμε τον αλγόριθμο bumping αντίστροφα. Έτσι, αν y είναι το στοιχείο του νέου κελιού του $T \leftarrow x$, τότε διαγράφουμε το κελί του y από το tableau και ψάχνουμε στην προηγούμενη γραμμή (από αυτή που βρίσκεται το y) *το δεξιότερο στοιχείο που είναι μικρότερο του* y, το αντικαθιστούμε με το y και επαναλαμβάνουμε, χρησιμοποιώντας κάθε φορά το στοιχείο που μόλις αντικαταστάθηκε, μέχρι να φτάσουμε στην πρώτη γραμμή οπότε καταλήγουμε στο αρχικό tableau T και σε ένα επιπλέον στοιχείο, που είναι το x.

Παράδειγμα: Έστω $T' = T \leftarrow 2$ το τελικό tableau του προηγούμενου παραδείγματος. Με την αντίστροφη διαδικασία θα ανακτήσουμε το tableau T καθώς και το στοιχείο $x = 2$ που εισήχθηκε αρχικά στο T. Με σκούρο χρώμα σημειώνεται το νέο κελί που προστέθηκε στο T.

$$\begin{array}{ccccc} 1 & 2 & 2 & 2 & 3 \\ 2 & 3 & 3 & & \\ 5 & 5 & & & \\ 6 & & & & \end{array} \qquad \begin{array}{ccccc} 1 & 2 & 2 & 2 & 3 \\ 2 & 3 & 3 & & \\ 5 & 6 & & & \end{array} \;\leftarrow 5 \qquad \begin{array}{ccccc} 1 & 2 & 2 & 2 & 3 \\ 2 & 3 & 5 & & \\ 5 & 6 & & & \end{array} \;\leftarrow 3$$

$$\begin{array}{ccccc} 1 & 2 & 2 & 3 & 3 \\ 2 & 3 & 5 & & \\ 5 & 6 & & & \end{array} \;\leftarrow 2$$

Το νέο κελί διαγράφεται από το διάγραμμα, το στοιχείο του (το 6) εκτοπίζει το δεξιότερο 5 της προηγούμενης (τρίτης) γραμμής, το εκτοπισμένο 5 εκτοπίζει το δεξιότερο 3 της προηγούμενης (δεύτερης) γραμμής και τέλος το εκτοπισμένο 3 εκτοπίζει το δεξιότερο 2 της προηγούμενης (πρώτης) γραμμής. Έτσι προκύπτει το T και το στοιχείο που εισήχθηκε αρχικά, δηλαδή το 2.

Γενικά, μπορούμε να εφαρμόσουμε τον αντίστροφο αλγόριθμο bumping σε ένα tableau για οποιοδήποτε κελί του, αρκεί αυτό να είναι εξωτερική γωνία του tableau. Αυτό γιατί κατά την εκτέλεση του αλγόριθμου bumping, το νέο κελί προκύπτει στο τέλος κάποιας γραμμής, είναι δηλαδή πάντα εξωτερική γωνία.

Κατά την εκτέλεση του αλγόριθμου bumping για την εισαγωγή ενός στοιχείου x σε ένα tableau T, προκύπτει μια μοναδική **διαδρομή εισαγωγής**. Η διαδρομή αυτή αποτελείται από όλα τα κελιά των οποίων τα στοιχεία μεταβλήθηκαν, καθώς και από το νέο κελί του tableau. Για παράδειγμα, για το tableau T του προηγούμενου παραδείγματος, η διαδρομή εισαγωγής φαίνεται με σκούρο χρώμα στο παρακάτω σχήμα.

$$T = \begin{array}{ccccc} 1 & 2 & 2 & 3 & 3 \\ 2 & 3 & 5 & & \\ 5 & 6 & & & \end{array}$$

$$T \leftarrow 2 = \begin{array}{ccccc} 1 & 2 & 2 & 2 & 3 \\ 2 & 3 & 3 & & \\ 5 & 5 & & & \\ 6 & & & & \end{array}$$

Για δύο κελιά B και B' με συντεταγμένες (i, j) και (k, l) αντίστοιχα, λέμε ότι το B είναι **αυστηρώς αριστερά (αντίστοιχα ασθενώς αριστερά)** του B', όταν $j < l$ (αντίστοιχα $j \leq l$). Ομοίως, λέμε ότι το B είναι **αυστηρώς κάτω (αντίστοιχα ασθενώς κάτω)** του B', όταν $k < i$ (αντίστοιχα $k \leq i$).

Για δύο διαδρομές εισαγωγής R και R', θα λέμε ότι η R είναι **αυστηρώς αριστερά (αντίστοιχα ασθενώς αριστερά)** της R', όταν σε κάθε γραμμή που περιέχει ένα κελί της R' υπάρχει ένα κελί της R που βρίσκεται αυστηρώς αριστερά (αντίστοιχα ασθενώς αριστερά) του κελιού της R'.

Μπορούμε να εφαρμόσουμε τον αλγόριθμο bumping για να ορίσουμε το **γινόμενο** δύο tableaux. Έστω δύο tableaux T και U. Το γινόμενό τους $\mathbf{T} \cdot \mathbf{U}$ προκύπτει εφαρμόζοντας κάθε φορά τον αλγόριθμο bumping, πραγματοποιώντας διαδοχικές εισαγωγές στοιχείων του U στο T έτσι ώστε το τελικό tableau να έχει αριθμό κελιών ίσο με το άθροισμα των αριθμών κελιών των T και U. Η σειρά με την οποία επιλέγουμε στοιχεία από το U για να τα εισάγουμε στο T, είναι από την τελευταία γραμμή προς τα πάνω και από το αριστερότερο στοιχείο προς το δεξιότερο κάθε γραμμής. Έτσι $T \cdot U = ((\cdots((T \leftarrow x_1) \leftarrow x_2)\cdots) \leftarrow x_{n-1}) \leftarrow x_n$, όπου $x_1, x_2, \ldots, x_n$ τα στοιχεία του U με τη σειρά που διαβάζονται από την τελευταία γραμμή προς τα πάνω και από την πρώτη στήλη προς τα δεξιά.

Παράδειγμα: Έστω τα tableaux

$$\mathrm{T} = \begin{ytableau} 1 & 2 & 2 & 3 & 3 \\ 2 & 3 & 5 \\ 5 & 6 \end{ytableau} \qquad\qquad \mathrm{U} = \begin{ytableau} 1 & 3 \\ 2 \end{ytableau}$$

και θέλουμε να υπολογίσουμε το $T \cdot U$. Διαβάζοντας τα στοιχεία του U από κάτω προς τα πάνω και από αριστερά προς τα δεξιά, έχουμε $T \cdot U = ((T \leftarrow 2) \leftarrow 1) \leftarrow 3$.

$$\mathrm{T{\cdot}U} = \begin{ytableau} 1 & 2 & 2 & 3 & 3 \\ 2 & 3 & 5 \\ 5 & 6 \end{ytableau} \cdot \begin{ytableau} 1 & 3 \\ 2 \end{ytableau} = \begin{ytableau} 1 & 2 & 2 & 2 & 3 \\ 2 & 3 & 3 \\ 5 & 5 \\ 6 \end{ytableau} \cdot \begin{ytableau} 1 & 3 \end{ytableau} =$$

$$\begin{ytableau} 1 & 1 & 2 & 2 & 3 \\ 2 & 2 & 3 \\ 3 & 5 \\ 5 \\ 6 \end{ytableau} \cdot \begin{ytableau} 3 \end{ytableau} = \begin{ytableau} 1 & 1 & 2 & 2 & 3 & 3 \\ 2 & 2 & 3 \\ 3 & 5 \\ 5 \\ 6 \end{ytableau}$$

Όπως μπορεί εύκολα να διαπιστωθεί με ένα αντιπαράδειγμα, η αντιμεταθετικότητα δεν ισχύει για την πράξη αυτή. Το ουδέτερο στοιχείο αυτής της πράξης είναι το κενό tableau, το οποίο θα συμβολίζουμε με $\emptyset$, έτσι ώστε

$$\emptyset \cdot T = T \cdot \emptyset = T, \text{ για κάθε tableau } T.$$

Ο αλγόριθμος sliding

Ο αλγόριθμος αυτός μετατρέπει ένα skew tableau σε ένα tableau.

Βήμα 1. Διάλεξε ένα κελί που να είναι εσωτερική γωνία του tableau.

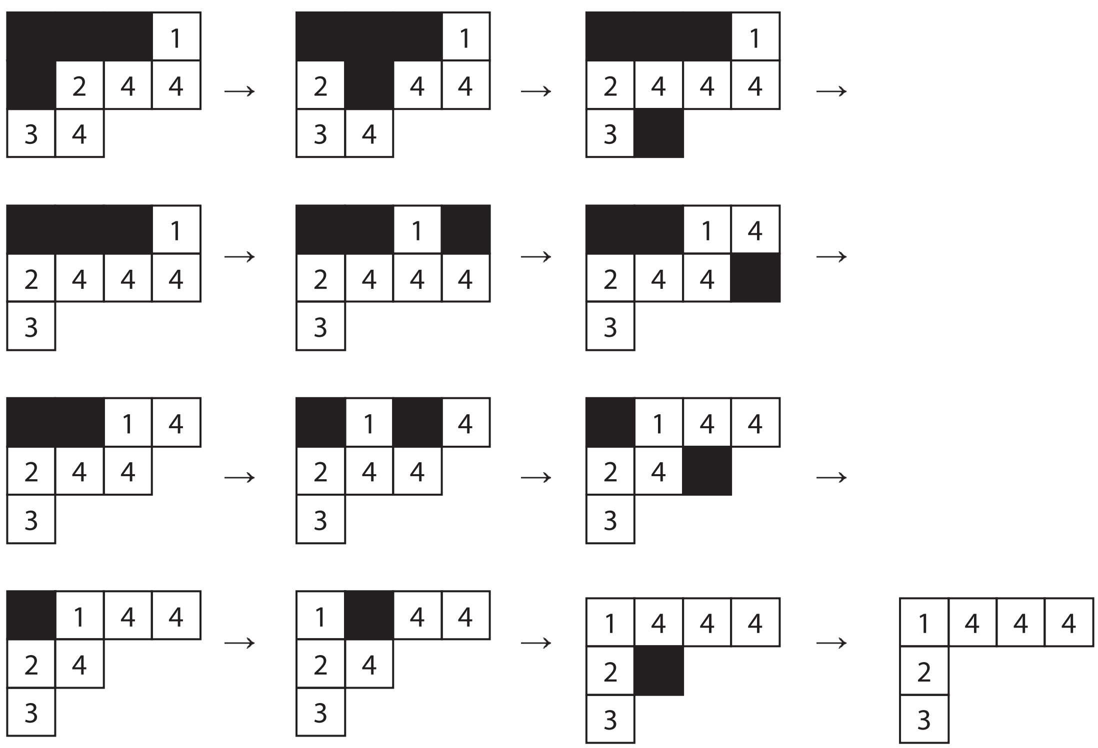
Βήμα 2. Αντιμετάθεσε το κελί αυτό με το μικρότερο από τα 2 κελιά που βρίσκονται δεξιά και κάτω από αυτό. Αν τα 2 κελιά είναι ίσα, τότε διάλεξε το κελί που είναι κάτω.[1]

Βήμα 3. Επανάλαβε το δεύτερο βήμα, μέχρις ότου το αρχικό κενό κελί (εσωτερική γωνία) γίνει εξωτερική γωνία οπότε και απαλείφεται από το διάγραμμα.

Βήμα 4. Επανάλαβε τα 3 παραπάνω βήματα, μέχρις ότου να μην υπάρχουν άλλες εσωτερικές γωνίες στο διάγραμμα.

Η σειρά με την οποία επιλέγουμε εσωτερικές γωνίες δεν έχει σημασία. Αν ονομάσουμε S το αρχικό skew tableau, τότε το tableau το οποίο προκύπτει ονομάζεται **rectification** του αρχικού skew tableau και συμβολίζεται $\mathbf{Rect}(\mathbf{S})$.

Παράδειγμα



[1] Σε περίπτωση που δεν υπάρχει κελί σε κάποια από τις δύο θέσεις (δεξιά και κάτω από το επιλεγμένο κελί), τότε το επιλεγμένο κελί αντιμετατίθεται με το κελί που υπάρχει δεξιά ή κάτω από αυτό.

Αν δίνονται δύο tableaux T και U, σχημάτων λ και μ αντίστοιχα, μπορούμε να σχηματίσουμε ένα skew tableau, το οποίο θα συμβολίζουμε με $T * U$, ως εξής:

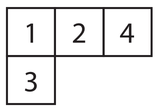
- Παίρνουμε ένα ορθογώνιο από κενά κελιά διαστάσεων $m \times n$ όπου m το πλήθος γραμμών του U και n το πλήθος στηλών του T.

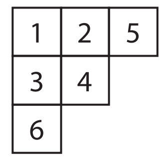
- Τοποθετούμε το U δεξιά από το κενό ορθογώνιο.

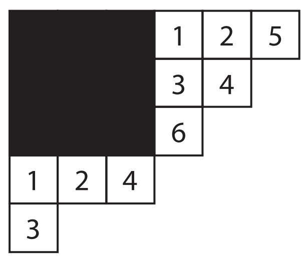
- Τοποθετούμε το T κάτω από το κενό ορθογώνιο.

Το διάγραμμα του $T * U$ θα το συμβολίζουμε με $[\lambda] * [\mu]$.

Παράδειγμα:

T = 

U = 

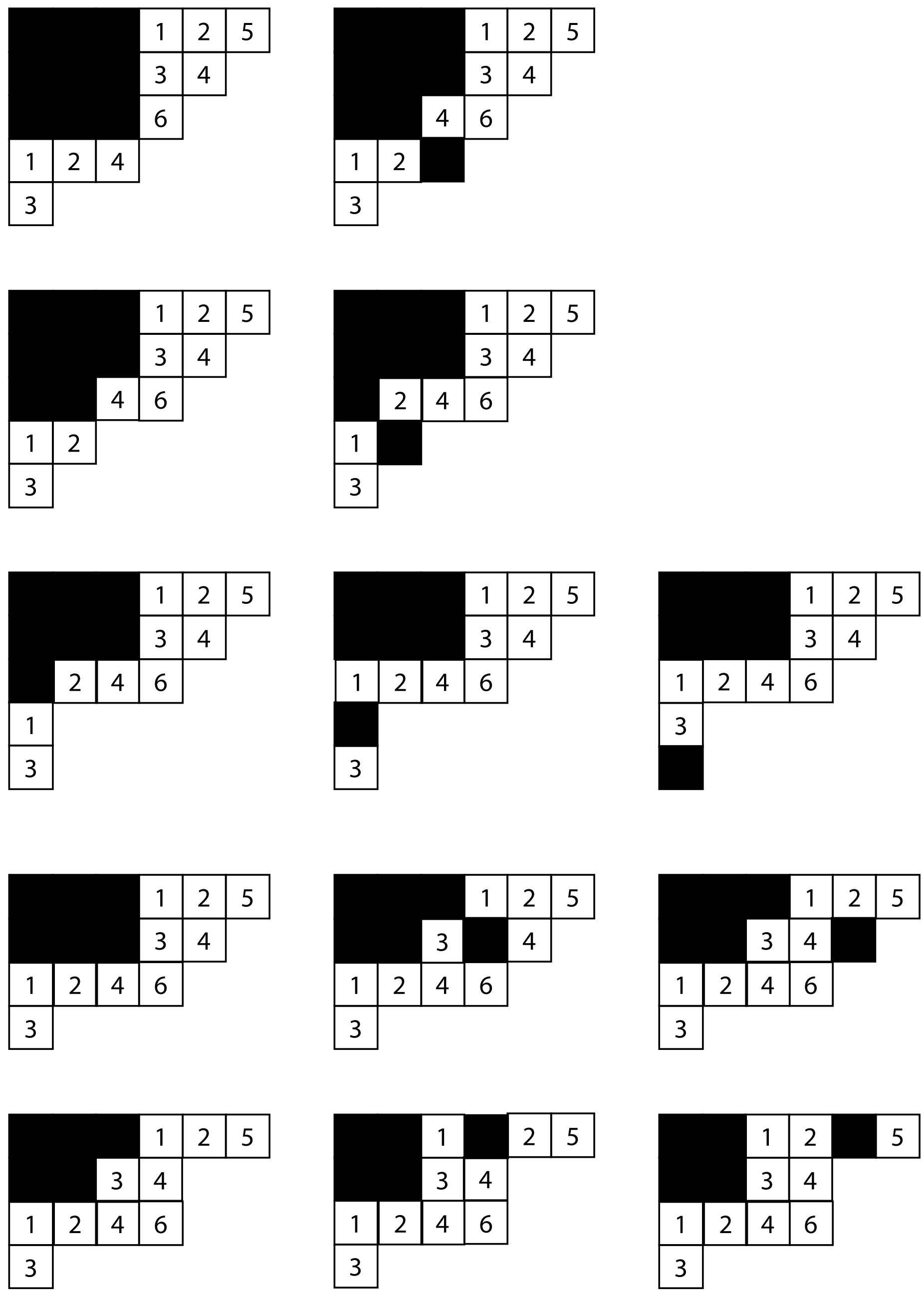
T*U = 

Αποδεικνύεται ότι το rectification του $T * U$ είναι ίσο με το $T \cdot U$, δηλαδή

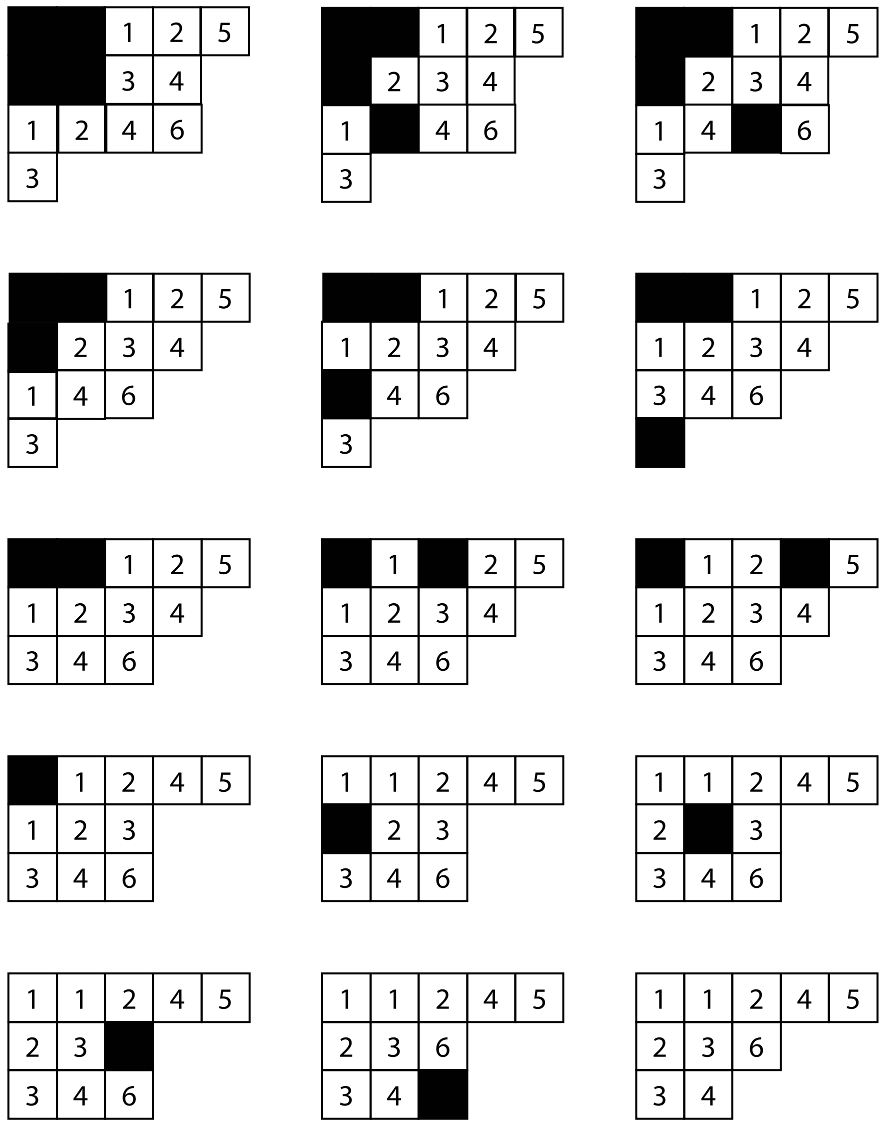
$$T \cdot U = Rect(T * U).$$

Έτσι, ο αλγόριθμος sliding μπορεί να εφαρμοστεί για να δώσει το ίδιο αποτέλεσμα με τον αλγόριθμο bumping, δηλαδή το γινόμενο $T \cdot U$ δύο tableaux.

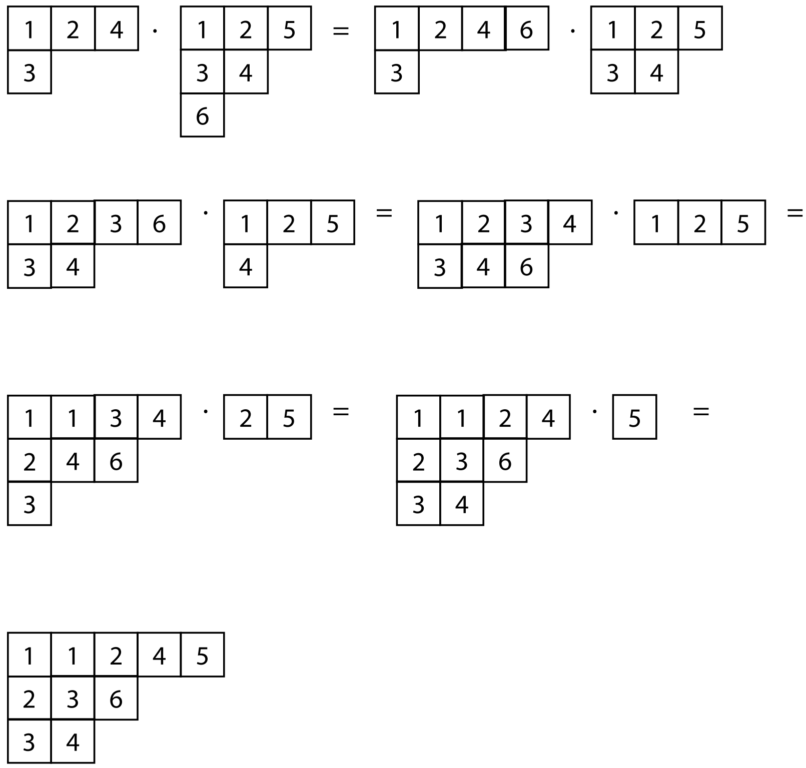
Παράδειγμα: Για τα T, U του προηγούμενου παραδείγματος, θα υπολογίσουμε τα $Rect(T * U)$ και $T \cdot U$ και θα επαληθεύσουμε ότι είναι ίσα. Δηλαδή, το ίδιο αποτέλεσμα μπορούμε να το πάρουμε είτε χρησιμοποιώντας τον αλγόριθμο bumping (για τη εύρεση του $T \cdot U$), είτε χρησιμοποιώντας τον αλγόριθμο sliding (για τη εύρεση του $Rect(T * U)$).



Σχήμα 4: Υπολογισμός του $Rect(T * U)$ με τον αλγόριθμο sliding.



Σχήμα 5: Υπολογισμός του $Rect(T * U)$ με τον αλγόριθμο sliding (συνέχεια).



Σχήμα 6: Υπολογισμός του $T \cdot U$ με τον αλγόριθμο bumping.

Απαρίθμηση των tableaux

Σε ένα διάγραμμα λ, ονομάζουμε **hook** ενός κελιού (i, j) και συμβολίζουμε με $H(i, j)$, το σύνολο όλων των κελιών που βρίσκονται στην ίδια γραμμή και δεξιά ή στην ίδια στήλη και κάτω από το κελί (i, j), καθώς και το ίδιο το κελί (i, j), δηλαδή,

$$H(i, j) = \{(i, j') : j' \geq j\} \cup \{(i', j) : i' \geq i\}.$$

Το **μήκος** $h(i, j)$ του hook είναι ίσο με το πλήθος των κελιών που ανήκουν στο hook του κελιού (i, j), δηλαδή,

$$h(i, j) = |H(i, j)|.$$

Πρόταση 1 (Hook length formula for SYT). *Το πλήθος f^λ των standard tableaux σχήματος λ, δίνεται από τον τύπο:*

$$f^\lambda = \frac{n!}{\prod_{(i,j)\in[\lambda]} h(i, j)}.$$

Ο τύπος αυτός αποδείχθηκε από τους Frame, Robinson και Thrall. Αργότερα ακολούθησαν αρκετές διαφορετικές αποδείξεις του τύπου αυτού κι από άλλους ερευνητές.

Ο αντίστοιχος τύπος για την απαρίθμηση των γενικευμένων tableaux έχει δοθεί από τον Stanley και είναι ο ακόλουθος:

Πρόταση 2 (Hook length formula for YT). *Το πλήθος $d_\lambda(m)$ των γενικευμένων tableaux σχήματος λ και με στοιχεία από το σύνολο $[m]$, δίνεται από τον τύπο:*

$$d_\lambda(m) = \prod_{(i,j)\in[\lambda]} \frac{m + j - i}{h(i, j)} = \frac{f^\lambda}{n!} \prod_{(i,j)\in[\lambda]} (m + j - i).$$

Αναπαράσταση των tableaux με λέξεις

Δοθέντος ενός tableau (ή skew tableau) T, ορίζουμε τη **λέξη tableau** και τη συμβολίζουμε με $w(T)$ ως την ακολουθία των στοιχείων του T αν αυτά διαβαστούν από κάτω προς τα πάνω και από αριστερά προς τα δεξιά. Για παράδειγμα, αν

$$T = \begin{array}{|c|c|c|c|c|c|} \hline 1 & 2 & 2 & 3 & 3 & 5 \\ \hline 2 & 3 & 5 & 5 \\ \cline{1-4} 4 & 4 & 6 & 6 \\ \cline{1-4} 5 & 6 \\ \cline{1-2} \end{array}$$

τότε $w(T) = 5644662355122335$.

Αν έχουμε τη λέξη tableau, τότε μπορούμε να κατασκευάσουμε το tableau διαβάζοντας τα γράμματα της λέξης από δεξιά προς τα αριστερά. Όταν συναντάμε ένα γράμμα μεγαλύτερο από αυτό που διαβάστηκε τελευταίο, τότε έχουμε αλλαγή γραμμής.

Κανονική διαδικασία κατασκευής ενός tableau από μια λέξη: Έστω η λέξη $w = w_1w_2\cdots w_n$. Ξεκινώντας με το κενό tableau $\emptyset$ και εφαρμόζοντας τον αλγόριθμο bumping για την εισαγωγή των στοιχείων της w στο tableau με σειρά από αριστερά προς τα δεξιά, προκύπτει το tableau

$$P(w) := \emptyset \leftarrow w := (\cdots((\emptyset \leftarrow w_1) \leftarrow w_2)\cdots) \leftarrow w_n.$$

Η παραπάνω διαδικασία ονομάζεται **κανονική διαδικασία** κατασκευής ενός tableau από μια λέξη. Το tableau που προκύπτει με αυτόν τον τρόπο από την λέξη w συμβολίζεται με $P(w)$.

Προφανώς το T μπορεί να προκύψει από περισσότερες από μια λέξεις με τα ίδια γράμματα αλλά σε διαφορετική σειρά. Μια από αυτές τις λέξεις είναι και η $w(T)$. Αντίθετα, μια λέξη αντιστοιχεί σε ένα μοναδικό tableau αφού η κανονική διαδικασία κατασκευής είναι μονοσήμαντη. Έτσι, η κανονική διαδικασία χωρίζει το σύνολο όλων των λέξεων σε κλάσεις ισοδυναμίας, με τις λέξεις που αντιστοιχούν στο ίδιο tableau να ανήκουν στην ίδια κλάση. Οι λέξεις αυτές ονομάζονται **ισοδύναμες κατά Knuth** ή K-ισοδύναμες.

Η αντιστοιχία Robinson-Schensted: Για κάθε λέξη w, συμβολίζουμε με $P(w)$ το μοναδικό tableau που προκύπτει από τη λέξη w με την κανονική διαδικασία που περιγράφτηκε νωρίτερα. Η λέξη tableau του $P(w)$ είναι ισοδύναμη κατά Knuth με την w. Όπως έχουμε δει, ο αλγόριθμος bumping για την εισαγωγή ενός στοιχείου μπορεί να αντιστραφεί, έτσι ώστε να προκύψει το αρχικό tableau και το στοιχείο που εισήχθηκε. Με το ίδιο σκεπτικό, μπορούμε να ανακτήσουμε μια λέξη w από το tableau $P(w)$ αρκεί να γνωρίζουμε τα στοιχεία που εισήχθηκαν και τη σειρά με την οποία έγινε η εισαγωγή τους. Για το λόγο αυτό, όταν κατασκευάζουμε το $P(w)$ από τη λέξη w, κατασκευάζουμε παράλληλα ένα δεύτερο tableau το οποίο συμβολίζεται με $Q(w)$ και αναπαριστά τη σειρά εισαγωγής των στοιχείων στο tableau. Το $Q(w)$ έχει το ίδιο σχήμα με το $P(w)$ και κάθε κελί του (i, j) περιέχει τον ακέραιο k όταν, στο $P(w)$, το κελί (i, j) δημιουργήθηκε στην k-οστή επανάληψη του αλγορίθμου.

Κατασκευή του ζεύγους tableaux (P, Q) **από μια λέξη** $w = w_1w_2\cdots w_r$

Βήμα 1. Θέσε $i = 1$ (μετρητής επαναλήψεων του αλγόριθμου), $P = Q = \emptyset$.

Βήμα 2. Θέσε $P = P \leftarrow w_i$. Αν το νέο κελί που προκύπτει στο P έχει συντεταγμένες (k, l), τότε πρόσθεσε στο Q ένα κελί στη θέση (k, l) που να περιέχει το στοιχείο i.

Βήμα 3. Αύξησε την τιμή του i κατά 1.

Βήμα 4. Αν $i < r + 1$, πήγαινε στο Βήμα 2.

Βήμα 5. Τερματισμός.

Ανάκτηση της λέξης w **από το ζεύγος tableaux** (P, Q)

Βήμα 1. Θέσε $i = r$, όπου r το πλήθος κελιών των P, Q. Θέσε $w = \emptyset$.

Βήμα 2. Αν (k, l) οι συντεταγμένες του κελιού με τη μεγαλύτερη τιμή (i) στο Q, τότε αφαίρεσέ το από το Q και εφάρμοσε αντίστροφο αλγόριθμο bumping για το κελί (k, l) στο P.

Βήμα 3. Πρόσθεσε το στοιχείο x που αφαιρείται από το P κατά τον αντίστροφο αλγόριθμο bumping, στη λέξη w από αριστερά έτσι ώστε $w = xw$.

Βήμα 4. Μείωσε την τιμή του i κατά 1.

Βήμα 5. Αν $i > 0$, πήγαινε στο Βήμα 2.

Βήμα 6. Τερματισμός.

Παράδειγμα: Έστω η λέξη $w = 5423123$. Εισάγουμε τα γράμματα της λέξης από αριστερά προς τα δεξιά στο $P(w)$ και ταυτόχρονα ενημερώνουμε το $Q(w)$. Στη συνέχεια παρουσιάζονται τα διαδοχικά ζευγάρια (P_i, Q_i) για κάθε επανάληψη i της διαδικασίας.

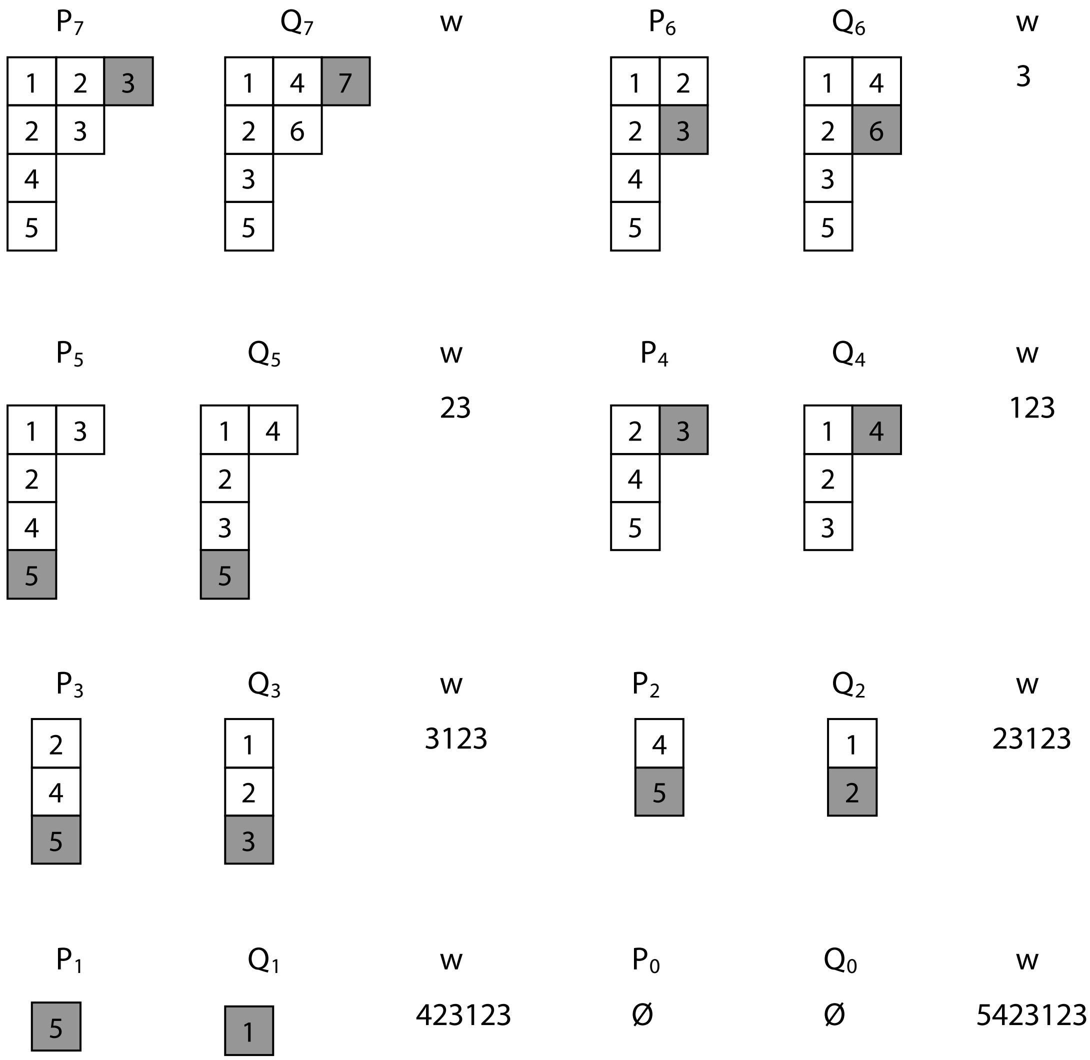
$$P_1 = \begin{array}{|c|}\hline 5\\\hline\end{array} \quad Q_1 = \begin{array}{|c|}\hline 1\\\hline\end{array} \quad P_2 = \begin{array}{|c|}\hline 4\\\hline 5\\\hline\end{array} \quad Q_2 = \begin{array}{|c|}\hline 1\\\hline 2\\\hline\end{array} \quad P_3 = \begin{array}{|c|}\hline 2\\\hline 4\\\hline 5\\\hline\end{array} \quad Q_3 = \begin{array}{|c|}\hline 1\\\hline 2\\\hline 3\\\hline\end{array}$$

$$P_4 = \begin{array}{|c|c|}\hline 2&3\\\hline \multicolumn{1}{|c|}{4}\\\cline{1-1} \multicolumn{1}{|c|}{5}\\\cline{1-1}\end{array} \quad Q_4 = \begin{array}{|c|c|}\hline 1&4\\\hline \multicolumn{1}{|c|}{2}\\\cline{1-1} \multicolumn{1}{|c|}{3}\\\cline{1-1}\end{array} \quad P_5 = \begin{array}{|c|c|}\hline 1&3\\\hline \multicolumn{1}{|c|}{2}\\\cline{1-1} \multicolumn{1}{|c|}{4}\\\cline{1-1} \multicolumn{1}{|c|}{5}\\\cline{1-1}\end{array} \quad Q_5 = \begin{array}{|c|c|}\hline 1&4\\\hline \multicolumn{1}{|c|}{2}\\\cline{1-1} \multicolumn{1}{|c|}{3}\\\cline{1-1} \multicolumn{1}{|c|}{5}\\\cline{1-1}\end{array} \quad P_6 = \begin{array}{|c|c|}\hline 1&2\\\hline 2&3\\\hline \multicolumn{1}{|c|}{4}\\\cline{1-1} \multicolumn{1}{|c|}{5}\\\cline{1-1}\end{array} \quad Q_6 = \begin{array}{|c|c|}\hline 1&4\\\hline 2&6\\\hline \multicolumn{1}{|c|}{3}\\\cline{1-1} \multicolumn{1}{|c|}{5}\\\cline{1-1}\end{array}$$

$$P_7 = \begin{array}{|c|c|c|}\hline 1&2&3\\\hline 2&\multicolumn{1}{c|}{3}\\\cline{1-2} \multicolumn{1}{|c|}{4}\\\cline{1-1} \multicolumn{1}{|c|}{5}\\\cline{1-1}\end{array} \quad Q_7 = \begin{array}{|c|c|c|}\hline 1&4&7\\\hline 2&\multicolumn{1}{c|}{6}\\\cline{1-2} \multicolumn{1}{|c|}{3}\\\cline{1-1} \multicolumn{1}{|c|}{5}\\\cline{1-1}\end{array}$$

Από το ζεύγος $(P(w), Q(w))$, μπορούμε να ανακτήσουμε τη λέξη w, αρκεί να εφαρμόσουμε τον αντίστροφο αλγόριθμο bumping με τη σειρά που υποδεικνύουν τα στοιχεία του Q, ξεκινώντας από το μεγαλύτερο. Έτσι, σε κάθε βήμα της διαδικασίας κατασκευάζουμε το ζεύγος (P_{i-1}, Q_{i-1}) από το ζεύγος (P_i, Q_i) και το στοιχείο που αφαιρείται από το P μετά την εφαρμογή του αντίστροφου αλγόριθμου bumping, είναι το i-οστό γράμμα της λέξης w.

Παράδειγμα: Έστω το ζεύγος tableaux (P, Q) του προηγούμενου παραδείγματος και θέλουμε τώρα να βρούμε τη λέξη w. Με σκούρο χρώμα σημειώνεται το μεγαλύτερο στοιχείο του Q, και το αντίστοιχο (που κατέχει την ίδια θέση) στοιχείο του P το οποίο πρόκειται να αφαιρεθεί από το P.



Το tableau $Q(w)$ είναι πάντα standard. Στην περίπτωση που η λέξη w είναι μετάθεση του $[n]$, τότε και το $P(w)$ είναι standard, οπότε προκύπτει μία αμφιμονοσήμαντη απεικόνιση $w \mapsto (P(w), Q(w))$ μεταξύ μεταθέσεων του $[n]$ και διατεταγμένων ζευγών από SYT ίδιου σχήματος με n στοιχεία.

Αύξουσες υπακολουθίες

Ο Schensted ανέπτυξε τον αλγόριθμο bumping κατά τη μελέτη του σχετικά με τα μήκη αύξουσων υπακολουθιών σε μια ακολουθία θετικών ακεραίων. Στα πλαίσια αυτής της μελέτης, κατέληξε σε πολύ σημαντικά αποτελέσματα που σχετίζουν τα tableaux με τα μήκη των υπακολουθιών αυτών.

Έστω μια λέξη (ακολουθία θετικών ακεραίων) $w = w_1 w_2 \cdots w_n$, με $w_i \in [m], i \in [n]$. Μια **αύξουσα υπακολουθία** (μήκους ℓ) στη λέξη w είναι μια ακολουθία στοιχείων $w_{k_1} \leq w_{k_2} \leq \cdots \leq w_{k_\ell}$, με $1 \leq k_1 < k_2 < \cdots < k_l \leq n$. Συμβολίζουμε με $L(w, 1)$ το μήκος της μέγιστης αύξουσας υπακολουθίας στη λέξη w.

Παράδειγμα: Έστω η λέξη $w = 122321134$. Παρακάτω φαίνονται οι δύο μέγιστες υπακολουθίες της w με μήκος 6.

$$\boxed{1} \quad \boxed{2} \quad \boxed{2} \quad \boxed{3} \quad 2 \quad 1 \quad 1 \quad \boxed{3} \quad \boxed{4}$$

$$\boxed{1} \quad \boxed{2} \quad \boxed{2} \quad 3 \quad \boxed{2} \quad 1 \quad 1 \quad \boxed{3} \quad \boxed{4}$$

Συμβολίζουμε με $L(w, k)$, $k \geq 1$, το μέγιστο αριθμό που μπορεί να επιτευχθεί ως το άθροισμα των μηκών k ξένων μεταξύ τους (χωρίς κοινά στοιχεία) αυξουσών υπακολουθιών στη w. Για τη λέξη $w = 122321134$ του προηγούμενου παραδείγματος έχουμε

$$\boxed{1} \quad \boxed{2} \quad \boxed{2} \quad \boxed{3} \quad 2 \quad 1 \quad 1 \quad \boxed{3} \quad \boxed{4} \qquad L(w,2) = 8$$

$$\boxed{1} \quad \boxed{2} \quad \boxed{2} \quad \boxed{3} \quad 2 \quad 1 \quad 1 \quad \boxed{3} \quad \boxed{4} \qquad L(w,3) = 9$$

Προφανώς $L(w, k) = 9$, για $k > 3$.

Ο αριθμός $L(w, k)$ μπορεί να προκύψει με περισσότερους από έναν συνδυασμούς υπακολουθιών, ανάλογα με τη λέξη. Για παράδειγμα, η λέξη $w' = 322111234$. έχει τα ίδια στοιχεία με τη w αλλά σε διαφορετική σειρά. Επίσης, η w' είναι λέξη tableau. Εύκολα μπορούμε να παρατηρήσουμε ότι $L(w, k) = L(w', k)$, για κάθε $k > 0$. Γενικά, ισχύει το ακόλουθο αποτέλεσμα:

Πρόταση 3. *Για δύο λέξεις w και w' ισχύει $L(w, k) = L(w', k)$, για κάθε $k > 0$ αν και μόνο αν οι w και w' είναι ισοδύναμες κατά Knuth.*

Παράδειγμα: Το tableau που αντιστοιχεί στη λέξη $w' = 322111234$ είναι το

$$P(w') = \begin{array}{|c|c|c|c|c|c|} \hline 1 & 1 & 1 & 2 & 3 & 4 \\ \hline 2 & 2 \\ \cline{1-2} 3 \\ \cline{1-1} \end{array}$$

Παρατηρούμε ότι ο αριθμός $L(w', k)$ είναι ίσος με το πλήθος των κελιών των k πρώτων γραμμών του $P(w')$. Πράγματι, ισχύει το ακόλουθο αποτέλεσμα.

Πρόταση 4. *Για κάθε λέξη w ισχύει*

$$L(w, k) = \lambda_1 + \lambda_2 + \cdots + \lambda_k, \text{για κάθε } k \in [r],$$

όπου $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_r)$ είναι η διαμέριση που ορίζει το διάγραμμα του tableau $P(w)$.

Έτσι, αν θέλουμε να υπολογίσουμε τα $L(w,k)$ για κάποια τυχαία λέξη w, αρκεί να κατασκευάσουμε το tableau $P(w)$ με την κανονική διαδικασία (εισάγοντας τα στοιχεία της w στο κενό tableau) και να μετρήσουμε το μήκος κάθε γραμμής του $P(w)$.

Παρατήρηση: Η τελευταία πρόταση μπορεί να χρησιμοποιηθεί και για την απαρίθμηση λέξεων με συγκεκριμένα χαρακτηριστικά. Για παράδειγμα, για να βρούμε το πλήθος των μεταθέσεων του [25] με μέγιστη αύξουσα υπακολουθία μήκους 21 και μέγιστη φθίνουσα υπακολουθία μήκους 3, τότε αρκεί να μετρήσουμε τα ζεύγη tableaux σχήματος $\lambda = (21,2,2)$ ή $\mu = (21,3,1)$, δηλαδή το ζητούμενο πλήθος ισούται με $(f^\lambda)^2 + (f^\mu)^2$, το οποίο υπολογίζεται με τη βοήθεια του τύπου hook length που είδαμε νωρίτερα.

Εύρεση μιας μέγιστης αύξουσας υπακολουθίας (ΜΑΥ) στην w. Μπορούμε να εντοπίσουμε μια αύξουσα υπακολουθία στην w μεγίστου μήκους $L(w,1) = \lambda_1$ ως εξής: Κάθε γράμμα w_i της w όταν εισάγεται στο tableau, αρχικά καταλαμβάνει ένα κελί της 1ης γραμμής (αργότερα κατά την εκτέλεση ίσως κάποιο άλλο γράμμα να το εκτοπίσει σε επόμενη γραμμή). Αν το κελί αυτό είναι το j-οστό της 1ης γραμμής, τότε το γράμμα w_i ονομάζεται j-**βασικό**. Η υπακολουθία της w που αποτελείται από όλα τα j-βασικά στοιχεία της είναι προφανώς γνησίως φθίνουσα για κάθε $j \in [\lambda_1]$ (γιατί κάθε στοιχείο της πλην του πρώτου εκτοπίζει ένα μεγαλύτερο j-βασικό στοιχείο που προηγείται στην υπακολουθία). Επομένως, η ΜΑΥ θα περιέχει το πολύ ένα j-βασικό στοιχείο για κάθε j. Επιπλέον, προφανώς για κάθε j-βασικό στοιχείο υπάρχει ένα $(j-1)$-βασικό στοιχείο αριστερά του στην w (αφού το $(j-1)$-οστό κελί δημιουργείται νωρίτερα από το j-οστό κελί). Επομένως, μπορούμε κατά την εκτέλεση του αλγορίθμου bumping να καταγράψουμε το j για κάθε γράμμα της w, κατασκευάζοντας έτσι μια ακολουθία που συμβολίζουμε με w^*, και η οποία ορίζεται από τη σχέση

$$w_i^* = j \Leftrightarrow \text{ το } w_i \text{ είναι } j\text{-βασικό}, \quad i \in [n],$$

και έπειτα διαβάζοντας την w από δεξιά προς τα αριστερά να επιλέξουμε το πρώτο λ_1-βασικό που συναντάμε, έπειτα το πρώτο $(\lambda_1 - 1)$-βασικό κ.ο.κ, επιλέγοντας έτσι συνολικά ένα στοιχείο για κάθε $j \in [\lambda_1]$. Πιο αυστηρά, θέτουμε $k_{\lambda_1+1} = n+1$ και για κάθε $i = \lambda_1, \lambda_1 - 1, \ldots, 1$, θέτουμε

$$k_i = \max\{j \in [n] : j < k_{i+1}, w_j^* = i\},$$

οπότε προκύπτει η ΜΑΥ $w_{k_1} w_{k_2} \cdots w_{k_{\lambda_1}}$.

Παράδειγμα: Για τη λέξη $w = (2,5,3,4,8,2,5,3,7,5)$, μήκους $n = 10$, εκτελώντας την κανονική διαδικασία κατασκευής του tableau

$$P(w) = \begin{array}{|c|c|c|c|c|}\hline 2 & 2 & 3 & 5 & 5 \\ \hline 3 & 4 & 7 \\ \cline{1-3} 5 & 8 \\ \cline{1-2}\end{array}$$

βρίσκουμε ότι $w^* = (1,2,2,3,4,2,4,3,5,5)$.

i	1	2	3	4	5	6	7	8	9	10
w_i	2	5	3	4	8	2	5	3	7	5
w_i^*	1	2	2	3	4	2	4	3	5	5

Κάθε μέγιστη αύξουσα υπακολουθία θα έχει μήκος $\lambda_1 = 5$, και μπορούμε να επιλέξουμε μια τέτοια, επιλέγοντας τις θέσεις (από δεξιά προς τα αριστερά) $10, 7, 4, 3, 1$, οπότε προκύπτει η υπακολουθία $w_1 w_3 w_4 w_7 w_{10} = (2,3,4,5,5)$.

Παρατηρήσεις: Επειδή στην κατασκευή αυτής της υπακολουθίας παίζει ρόλο μόνο η πρώτη γραμμή του tableau, μπορούμε να κατασκευάσουμε μόνο αυτή. Αν χρησιμοποιήσουμε δυαδική αναζήτηση για την εισαγωγή των στοιχείων στην γραμμή, τότε ο αλγόριθμος κατασκευής της μέγιστης αύξουσας υπακολουθίας θα είναι χρόνου $O(n \log n)$ και αυτός είναι ο ταχύτερος γνωστός αλγόριθμος εύρεσης ΜΑΥ. Ομοίως, μπορούμε να βρούμε μια μέγιστη φθίνουσα υπακολουθία, αν αντιστρέψουμε τη σειρά των γραμμάτων της w. Σημειώνεται ότι όταν η w δεν έχει επαναλαμβανόμενα γράμματα (δηλαδή είναι μετάθεση), τότε αντιστρέφοντας την w θα πάρουμε το ανάστροφο tableau του $P(w)$, οπότε το μήκος της μέγιστης φθίνουσας υπακολουθίας της w ισούται με το πλήθος γραμμών του $P(w)$.

Κατασκευή όλων των ΜΑΥ της ακολουθίας w. Έστω w μια ακολουθία θετικών ακεραίων μήκους n και έστω w^* η ακολουθία με $w^*_i = j$ ανν το w_i είναι j-βασικό, για κάθε $i \in [n]$. Ως γνωστό (από τα προηγούμενα), είναι $w^* \in [\lambda_1]^n$, όπου λ_1 το μήκος της πρώτης γραμμής του tableau $P(w)$, και κάθε αύξουσα υπακολουθία μεγίστου μήκους (ΜΑΥ) της w έχει μήκος λ_1. Είναι επίσης γνωστό (από τα προηγούμενα) ότι αν k είναι μια (γνησίως αύξουσα) ακολουθία δεικτών στο $[n]$ τέτοια ώστε η $w \circ k$ να είναι ΜΑΥ της w, τότε η $w \circ k$ περιέχει ακριβώς ένα j-βασικό στοιχείο για κάθε $j \in [\lambda_1]$, δηλαδή η $w^* \circ k$ είναι ένα προς ένα και επί του $[\lambda_1]$.

Θα δειχθεί ότι η $w^* \circ k$ είναι και γνησίως αύξουσα, οπότε θα είναι $w^* \circ k = (1, 2, \dots, \lambda_1)$ για κάθε k. Πράγματι, αν υποτεθεί το αντίθετο, τότε υπάρχουν δείκτες s, t, με $1 \leq s < t \leq n$, $w_s \leq w_t$ και $w^*_s \geq w^*_t$. Το w_t εισάγεται αφού έχει εισαχθεί το w_s και στο κελί a_t της 1ης γραμμής, το οποίο ήδη περιέχει κάποιο στοιχείο έστω $w_{s'}$, με $s' < t$, διότι $w^*_s \geq w^*_t$. Επομένως, είναι $w_t < w_{s'}$ (αφού το w_t εκτοπίζει το $w_{s'}$) και $w_{s'} \leq w_s$ (αφού η 1η γραμμή του $P(w)$ είναι σε αύξουσα σειρά), δηλαδή $w_t < w_{s'} \leq w_s$, άτοπο.

Επομένως, όταν η $w \circ k$ είναι ΜΑΥ, τότε

$$w^* \circ k = (1, 2, \dots, \lambda_1).$$

Ωστόσο, το αντίστροφο δεν ισχύει πάντα. Παρόλα αυτά, η ιδιότητα αυτή μας επιτρέπει να διατυπώσουμε έναν BEST αλγόριθμο backtracking για την εύρεση όλων των ΜΑΥ, ως εξής: Ορίζουμε C_i να είναι το choice set του όρου w_i, $i \in [n]$, με

$$C_i = \begin{cases} \{j \in [n] :, j > i, w_j \geq w_i, w^*_j = w^*_i + 1, C_j \neq \emptyset\}, & w^*_i < \lambda_1, \\ \{n+1\}, & w^*_i = \lambda_1, \end{cases}$$

δηλαδή το σύνολο των δεικτών $j > i$ για τους οποίους το w_j επεκτείνει μια υπακολουθία με τελευταίο όρο w_i. Για το w_j θα πρέπει να είναι $w^*_j = w^*_i + 1$, βάσει της προηγούμενης ιδιότητας, αλλιώς η επέκταση αυτή δεν θα οδηγήσει τελικά σε ΜΑΥ. Υπολογίζουμε τα C_i με δυναμικό προγραμματισμό, από $i = n$ έως 1, οπότε όταν το C_i προκύπτει κενό, αυτό σηματοδοτεί ένα i που δεν μπορεί να οδηγήσει σε επέκταση άρα ούτε και σε λύση. Για τον λόγο αυτόν, ειδικά στην περίπτωση που $w^*_i = \lambda_1$, τότε το C_i ορίζεται όχι ως κενό αλλά ως το μονοσύνολο $\{n+1\}$, ώστε το i αυτό να μην θεωρηθεί ως αδιέξοδο.

```
w  = [9, 10, 19, 13, 12, 3, 17, 2, 14, 13, 12, 3, 5, 10, 15, 19, 17, 18, 13, 2]
ws = [1,  2,  3,  3,  3, 1,  4, 1,  4,  4,  4, 2, 3,  4,  5,  6,  6,  7,  5, 2]

Generating all longest non-decreasing subsequences of w...
[2,  3,  5, 10, 15, 17, 18]
[3,  3,  5, 10, 15, 17, 18]
[9, 10, 12, 12, 15, 17, 18]
[9, 10, 12, 13, 15, 17, 18]
[9, 10, 12, 14, 15, 17, 18]
[9, 10, 13, 13, 15, 17, 18]
[9, 10, 13, 14, 15, 17, 18]
7 longest non-decreasing subsequences found.
```