Αλγόριθμος Z

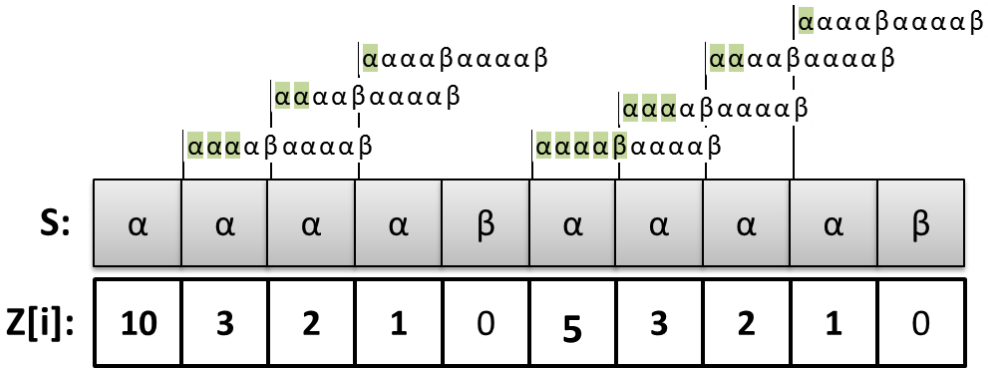
- Τι είναι + παράδειγμα
- Γιατί να θέλουμε να τον υπολογίσουμε + παράδειγμα
- Γιατί να μην κάνουμε brute force + παράδειγμα για τετραγωνική πολυπλοκότητα
- Γρήγορος Αλγόριθμος
- Γιατί είναι γρήγορος
- Υλοποίηση
- Ερωτήσεις κατανόησης
- Προβλήματα

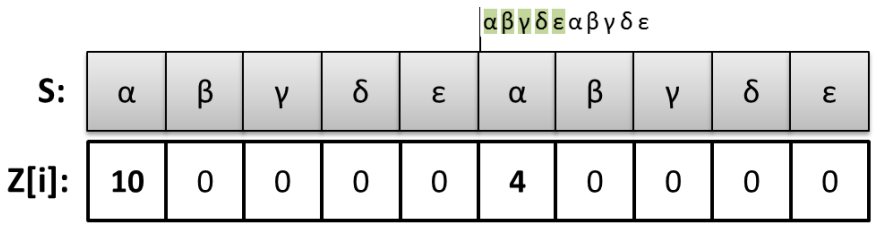
Τι είναι;

Ο Αλγόριθμος Z είναι ο υπολογισμός κάποιων τιμών για μία συμβολοσειρά που μας επιτρέπει να κάνουμε διάφορα ενδιαφέροντα πράγματα (όπως string searching) γρήγορα. Πιο συγκεκριμένα για μία συμβολοσειρά S

Z [i] = μήκος μεγαλύτερης ακολουθίας που ξεκινάει από το i και είναι prefix της S

Ας δούμε κάποια παραδείγματα:



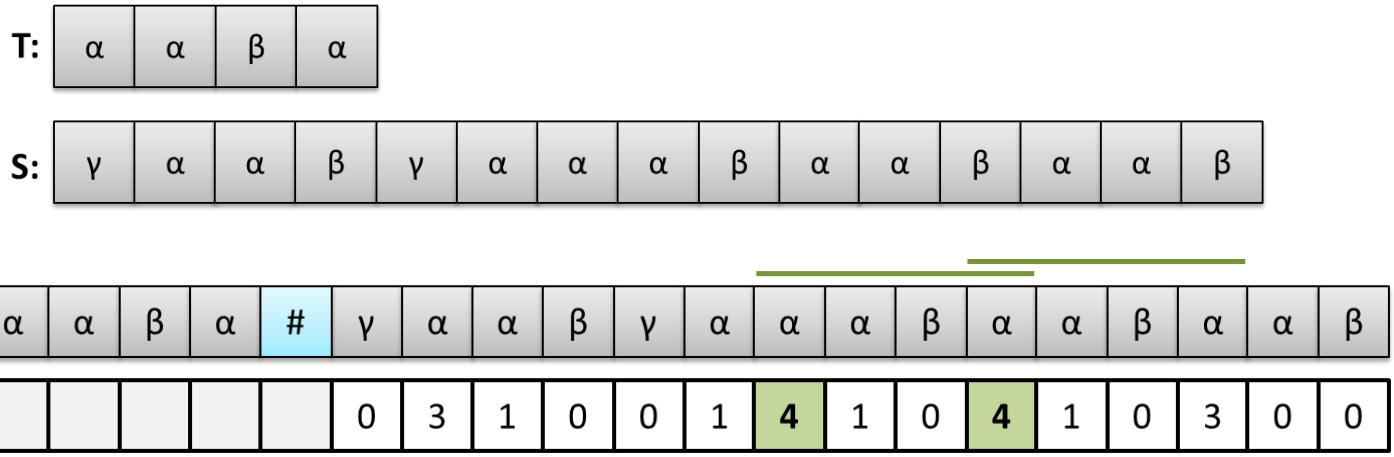


Γιατί να θέλουμε να υπολογίσουμε αυτές τις τιμές Z;

Η πιο κοινή χρήση αυτού του πίνακα είναι για την εύρεση μίας συμβολοσειράς T μέσα σε μία άλλη S σε γραμμικό χρόνο. Αφιερώστε λίγο χρόνο να σκεφτείτε πώς θα μπορούσαμε να βρούμε όλα τα σημεία που ξεκινάει μία συμβολοσειρά T στο S.

Αυτό που πρέπει να κάνουμε είναι να κατασκευάσουμε μία συμβολοσειρά ώστε οι τιμές Z να μας λένε πόσο μεγάλο είναι το prefix του T που ξεκινάει σε αυτό το σημείο. Η συμβολοσειρά που το κάνει αυτό είναι η $T+'\#'+S$. Έτσι, οι τιμές του Z που αφορούν το μέρος του S θα μας λένε ποιο είναι το μεγαλύτερο prefix του T που ξεκινάει από ένα σημείο στο S. Άμα το $Z[i] = |T|$ τότε σημαίνει ότι από εκεί ξεκινάει μία συμβολοσειρά T.

Για παράδειγμα

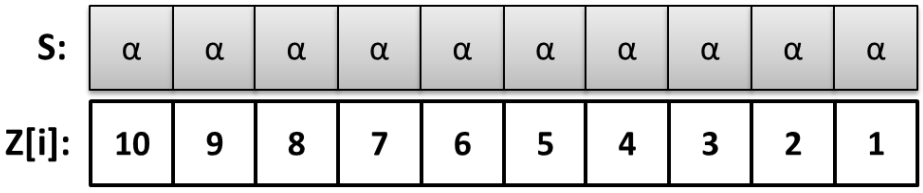


Brute Force Λύση

Ο πιο εύκολος τρόπος να λύσουμε αυτό το πρόβλημα θα ήταν να ξεκινήσουμε από κάθε i και να αυξάνουμε έναν counter ώσπου το επόμενο στοιχείο στη συμβολοσειρά είναι διαφορετικό από το επόμενο στοιχείο στο prefix. Η λογική αυτή φαίνεται καλύτερα στον παρακάτω κώδικα.

```
for (int i = 1; i < n; ++i) {
      z[i] = 0;
      while(i + z[i] < n && s[z[i]] == s[i + z[i]]) z[i]++;
}
```

Αυτός ο αλγόριθμος θα κάνει συνολικά $Z[1] + Z[2] + \cdots + Z[n-1]$ συγκρίσεις. Στην χειρότερη περίπτωση (δηλαδή η περίπτωση που δίνει τις μεγαλύτερες τιμές για κάθε $Z[i]$[1]) είναι η ακόλουθη (στη γενική της περίπτωση με n α):



Η πολυπλοκότητα του αλγόριθμου επομένως είναι $O(n + (n-1) + \cdots + 2 + 1) = O\left(\frac{1}{2} \cdot n \cdot (n-1)\right) = O(n^2)$

[1] Μπορείτε να αιτιολογήσετε είναι πραγματικά οι μεγαλύτερες τιμές για κάθε $Z[i]$;

Μπορούμε να κάνουμε κάτι πιο γρήγορο;

Ναι, εννοείται. Αυτό που πρέπει να κοιτάξουμε είναι σε ποιο σημείο υπολογίζουμε πράγματα που θα μπορούσαμε να είχαμε βρει από αυτά που ήδη ξέρουμε. Ας δούμε μία γενίκευση του 2ου παραδείγματος:

S:	α	α	α	α	β	α	α	α	α	β	α	α	α	α	β	α	α	α	α	β
Z[i]:	**20**	**3**	**2**	**1**	0	**15**	**3**	**2**	**1**	0	**10**	**3**	**2**	**1**	0	**5**	**3**	**2**	**1**	0

Μήπως μπορούσαμε να έχουμε προβλέψει το μοτίβο 3, 2, 1 γνωρίζοντας τις τιμές 15, 10 και 5? Η τιμή 15 μας λέει ότι οι παρακάτω συμβολοσειρές είναι ίσες:

α	α	α	α	β	α	α	α	α	β	α	α	α	α	β	α	α	α	α	β
α	α	α	α	β	α	α	α	α	β	α	α	α	α	β					

Ας συγκρίνουμε τις τιμές Z των δύο ακολουθιών:

20	**3**	**2**	**1**	0	**15**	**3**	**2**	**1**	0	**10**	**3**	**2**	**1**	0	**5**	**3**	**2**	**1**	0
15	**3**	**2**	**1**	0	**10**	**3**	**2**	**1**	0	**5**	**3**	**2**	**1**	0					

Ισχύει ότι οι τιμές Z των δύο ακολουθιών είναι ίδιες;
Όχι. Αυτό δεν συμβαίνει για το 15 ούτε για το 10 και το 5.

Τότε μήπως ισχύει ότι οι τιμές που έχουν $i + Z[i]$ μικρότερο από 15 θα είναι ίσες;
Ναι, αλλά ας δούμε γιατί. Το 15 που έχουμε υπολογίσει στην αρχή μας ορίζει όλα τα στοιχεία από το 5 έως το 19. Επομένως αυτά θα είναι ίσα με τους πρώτους 15 χαρακτήρες του S. Επομένως αν μία συμβολοσειρά prefix ξεκινάει μέσα σε αυτούς τους 15 χαρακτήρες και τελειώνει μέσα σε αυτούς (δηλαδή υπάρχει στοιχείο που διαφέρει από το επόμενο στο prefix) τότε θα διαφέρει και στο αντίστοιχο σημείο στους χαρακτήρες 5 έως 19.

Ας δούμε για παράδειγμα τι γίνεται με την τιμή $Z[1]$:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
α	α	α	α	β	α	α	α	α	β	α	α	α	α	β	α	α	α	α	β

Ο λόγος που σταματάει είναι ότι το **β** δεν είναι ίσο με **α** και έτσι γίνονται matched μόνο οι πρώτοι τρεις χαρακτήρες. Το γεγονός όμως ότι είναι ίσοι οι πρώτοι 15 χαρακτήρες μας λέει ότι το **β** θα υπάρχει και στην επόμενη ακολουθία επομένως:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
α	α	α	α	β	α	α	α	α	β	α	α	α	α	β	α	α	α	α	β

Επομένως, μπορούμε να γενικεύσουμε αυτή τη λογική κρατώντας το δεξιότερο διάστημα $[L, R]$ που ξέρουμε ότι έχει γίνει matched με το prefix, όταν έχουμε $Z[i - L] + i < R$ τότε ξέρουμε ότι το $Z[i] = Z[i - L]$

Μπορούμε να κάνουμε κάτι για τις υπόλοιπες τιμές;
Αυτό που ξέρουμε είναι ότι κάνουν match έως το τέλος τις περιοχής που έχουμε εξερευνήσει. Άρα μπορούμε να ξεκινήσουμε ένα νέο διάστημα $[L, R]$ και να το επεκτείνουμε. Έτσι θα βρούμε την τιμή του $Z[i]$ και το καινούργιο διάστημα. Το καλό μέρος εδώ είναι ότι το R μόνο αυξάνει επομένως στο σύνολο δεν θα γίνουν πάνω από $|S|$ αυξήσεις.

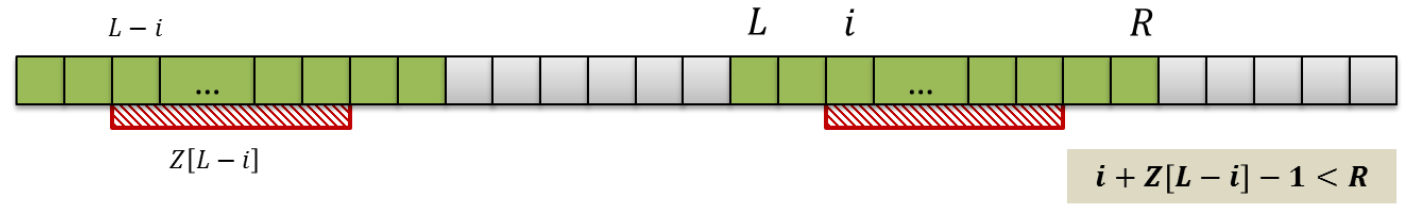
Ο Αλγόριθμος

Τώρα που είδαμε περίπου την λογική πίσω από τον αλγόριθμο μπορούμε να δούμε αναλυτικά τις περιπτώσεις. Σε κάθε βήμα του αλγορίθμου θα διατηρούμε το δεξιότερο διάστημα[2] $[L, R]$ και θα υπολογίζουμε το $Z[i]$.

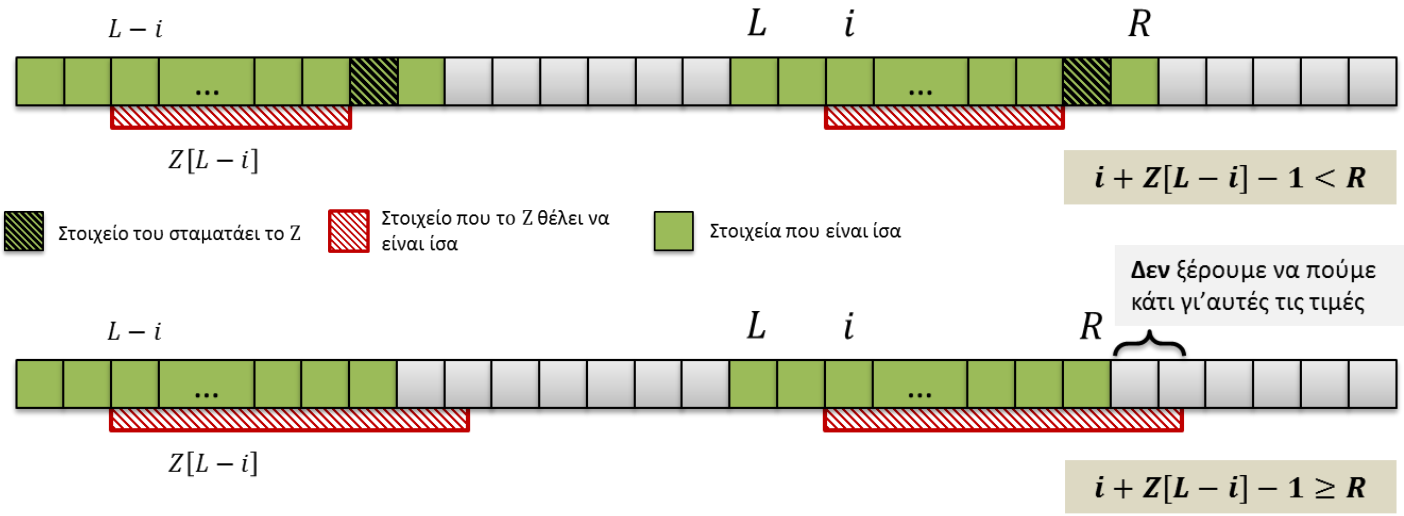
- *Αν το $i > R$, δηλαδή το index μας βρίσκεται έξω από αυτά που έχουμε ήδη κοιτάξει*
  Τότε θα χρησιμοποιήσουμε την brute force λογική για να υπολογίσουμε το $Z[i]$ και στο τέλος θα αναβαθμίσουμε το διάστημα σε $[i, Z[i] + i - 1]$

- *Αν το $i \leq R$, δηλαδή το index βρίσκεται μέσα σε αυτά που έχουμε ήδη κοιτάξει*
  Τότε μπορούμε να βάλουμε την τιμή του $Z[i] = Z[i - L]$ δηλαδή της τιμής που έχουμε ήδη υπολογίσει. Αλλά όπως είπαμε αυτή η τιμή μπορεί να είναι πολύ μεγάλη (εννοώντας ότι $Z[i - L] + i > R$ [3]) επομένως θα πρέπει να πάρουμε $Z[i] = \min(Z[i - L],\ R - i + 1)$. Τώρα αν το $i + Z[i]$ είναι ίσο με το R πρέπει να ελέγξουμε τι γίνεται με τις υπόλοιπες τιμές και το κάνουμε αυτό πάλι με την brute force λογική. Επομένως αν χρειαστεί να αυξήσουμε το R θα πρέπει να αναβαθμίσουμε και το διάστημα μας.

Διαγραμματικά μπορούμε να δούμε την δεύτερη περίπτωση ως δύο υποπεριπτώσεις:



Μπορούμε να μιλήσουμε με σιγουριά ότι οι τιμή του $Z[i]$ είναι $Z[L - i]$ γιατί στις τιμές που γνωρίζουμε ότι είναι ίδιες στο διάστημα $[L, R]$ εμφανίζεται και το στοιχείο που φράζει το $Z[L - i]$ και επομένως φράζει και το $Z[i]$.



[2] Αυτό που έχει το μεγαλύτερο δεξί endpoint.
[3] Γιατί δεν είναι φυσιολογικό αυτό;

Γιατί είναι γρήγορος ο αλγόριθμος[4]*;*
Το μέρος του αλγορίθμου που παίρνει τον έξτρα χρόνο είναι το κομμάτι που τρέχουμε την brute force λογική. Αλλά κάθε φορά που το κάνουμε αυτό αυξάνουμε και το R. Το R μπορεί να πάρει το πολύ $|S|$ τιμές επομένως ο αλγόριθμος τρέχει σε $O(|S|)$.

Γιατί είναι σωστός ο αλγόριθμος;
Κάθε φορά που σταματάμε την brute force μέθοδο τότε σημαίνει ότι έχουμε βρει το χαρακτήρα (ή το τέλος) που σταματάει το prefix που ξεκινάει από την θέση i και ότι όλα τα προηγούμενα είναι στοιχεία είναι ίσα.

Υλοποίηση Αλγόριθμου

Η υλοποίηση είναι σχετικά μικρή και απλή

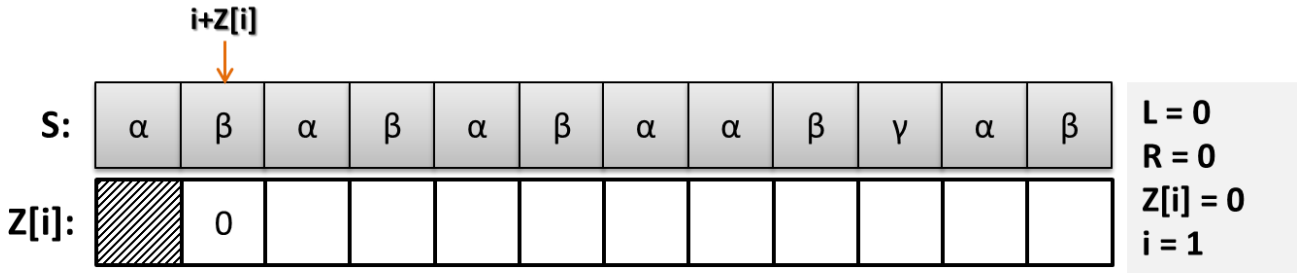
```
int L = 0, R = 0;
for (int i = 1; i < n; ++i) {
      Z[i] = 0;
      // Το στοιχείο μας είναι στο διάστημα που έχουμε ελέγξει
      if (i <= R)
            Z[i] = min(R - i + 1, Z[i - L]);
      while (i + Z[i] < n && S[Z[i]] == S[i + Z[i]]) Z[i]++;
      // Αν έχουμε καινούργιο δεξιότερο διάστημα
      if (i + Z[i] - 1 > R) {
            // Αναβάθμιση του διαστήματος
            l = i; r = i + Z[i] - 1;
      }
}
```

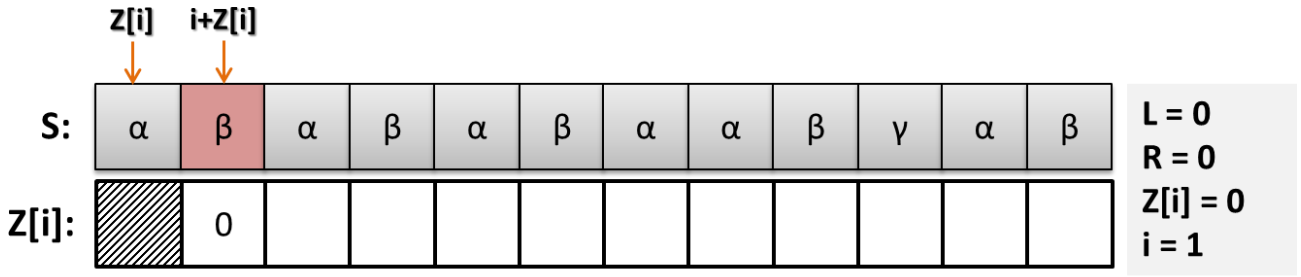
Σχόλια:

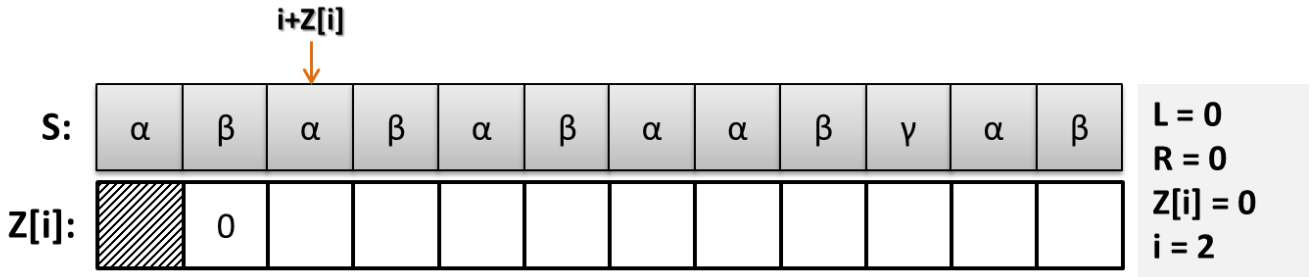
- Υπάρχουν αρκετές διαφορετικές παραλλαγές αυτού του αλγορίθμου
- Θα μπορούσαμε να αφαιρέσουμε τον έλεγχο $i + Z[i] < n$, αν θέσουμε τον τελευταίο χαρακτήρα $S[n] =' \#'$.
- Παρατηρήστε ότι δεν υπολογίζουμε την τιμή του $Z[0]$

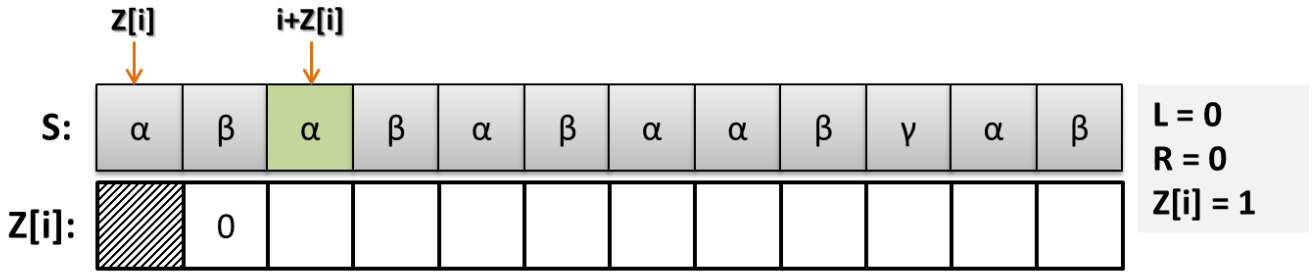
[4] Μπορεί να είναι πιο εύκολο να κατανοήσετε την πολυπλοκότητα του αλγορίθμου αν κοιτάξετε τον κώδικα πρώτα.

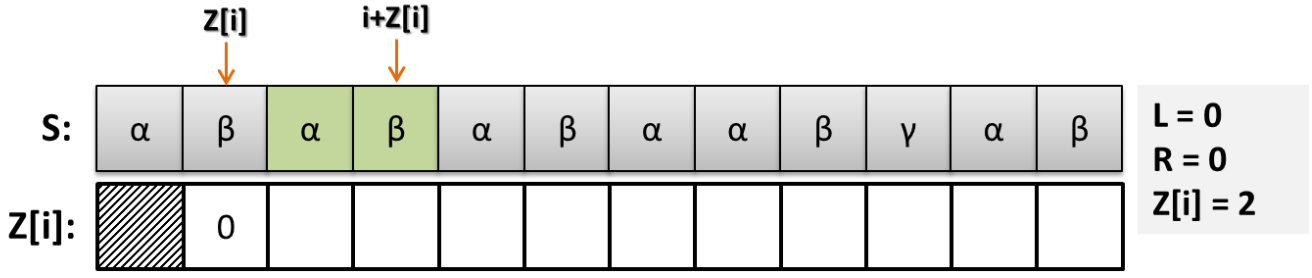
Παραδείγματα Εκτέλεσης

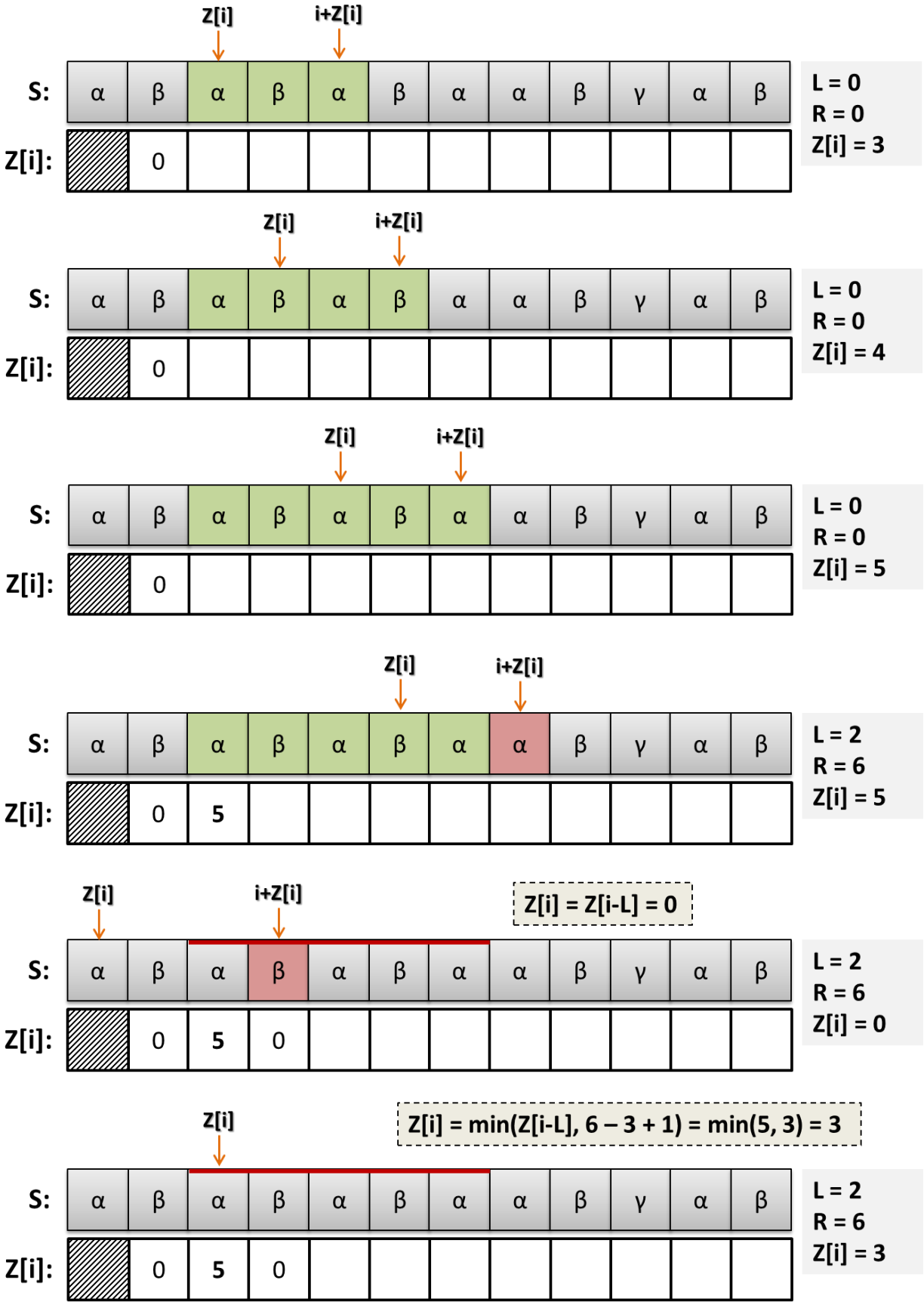


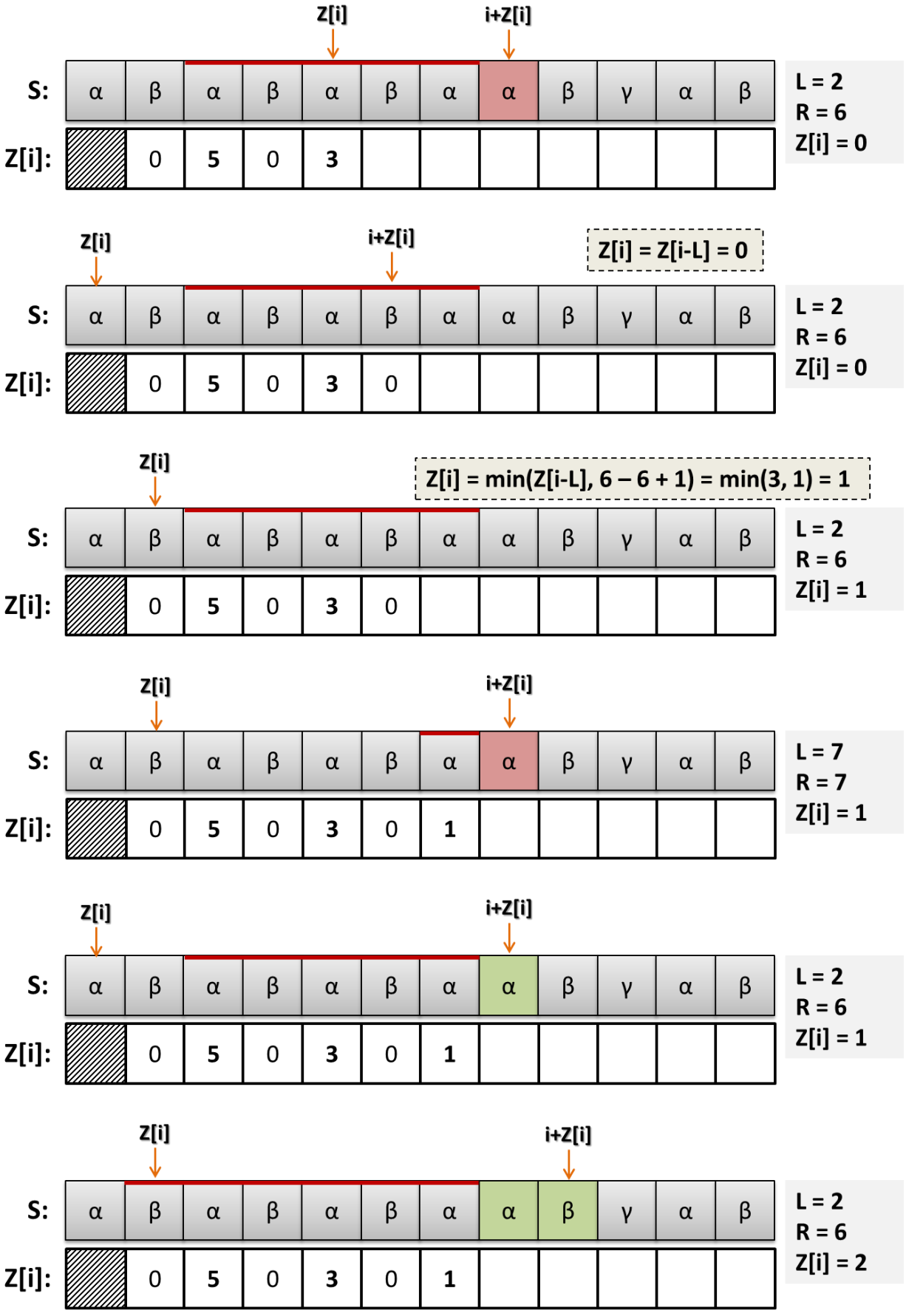


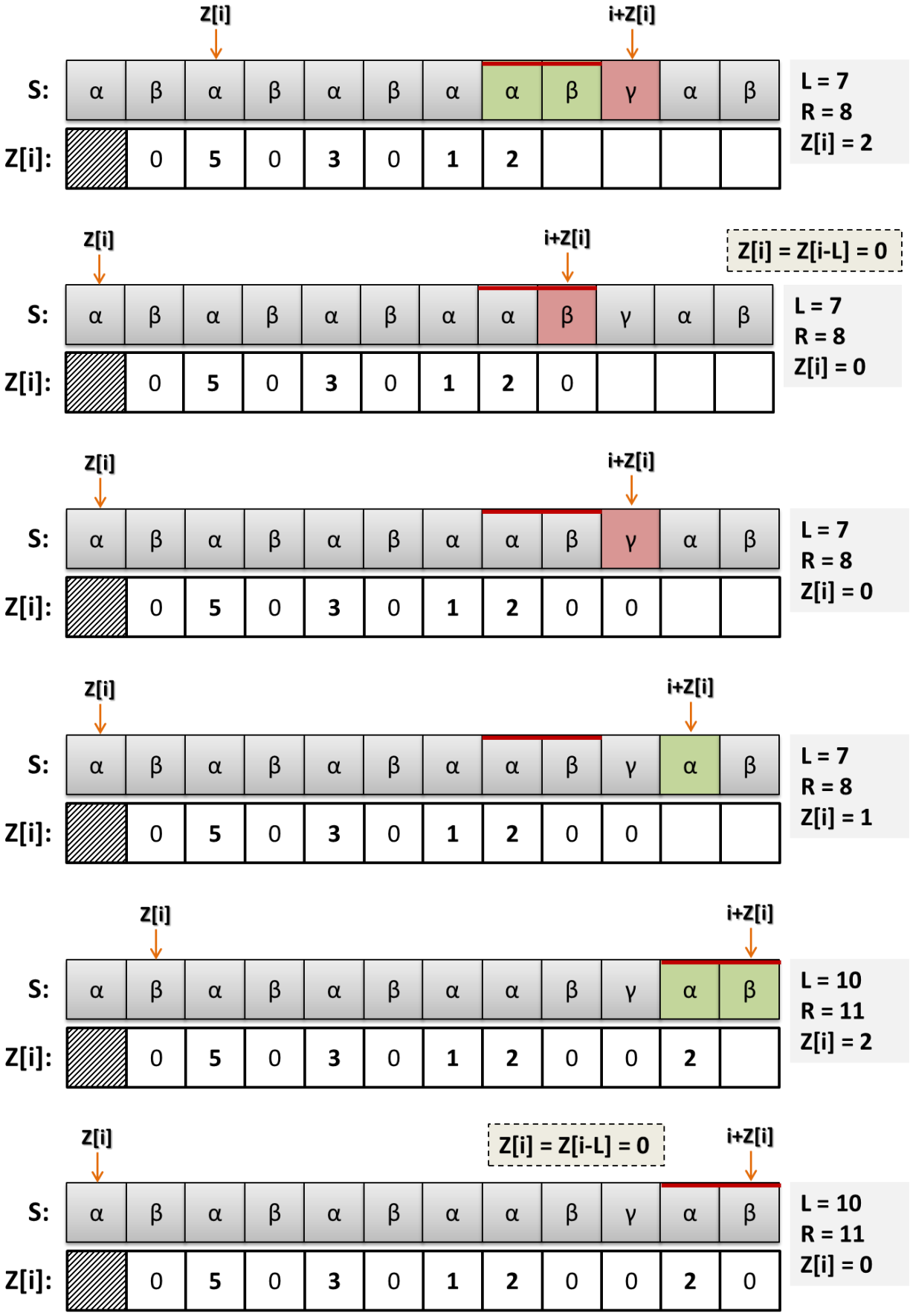







Z[i]
i+Z[i]
S:
α β α β α β α α β γ α β
Z[i]:
0
L = 0
R = 0
Z[i] = 3
Z[i]
i+Z[i]
S:
α β α β α β α α β γ α β
Z[i]:
0
L = 0
R = 0
Z[i] = 4
Z[i]
i+Z[i]
S:
α β α β α β α α β γ α β
Z[i]:
0
L = 0
R = 0
Z[i] = 5
Z[i]
i+Z[i]
S:
α β α β α β α α β γ α β
Z[i]:
0
5
L = 2
R = 6
Z[i] = 5
Z[i]
i+Z[i]
Z[i] = Z[i-L] = 0
S:
α β α β α β α α β γ α β
Z[i]:
0
5
0
L = 2
R = 6
Z[i] = 0
Z[i]
Z[i] = min(Z[i-L], 6 – 3 + 1) = min(5, 3) = 3
S:
α β α β α β α α β γ α β
Z[i]:
0
5
0
L = 2
R = 6
Z[i] = 3


Z[i]
i+Z[i]
S:
α β α β α β α α β γ α β
Z[i]:
0 5 0 3
L = 2
R = 6
Z[i] = 0
Z[i]
i+Z[i]
Z[i] = Z[i-L] = 0
S:
α β α β α β α α β γ α β
Z[i]:
0 5 0 3 0
L = 2
R = 6
Z[i] = 0
Z[i]
Z[i] = min(Z[i-L], 6 – 6 + 1) = min(3, 1) = 1
S:
α β α β α β α α β γ α β
Z[i]:
0 5 0 3 0
L = 2
R = 6
Z[i] = 1
Z[i]
i+Z[i]
S:
α β α β α β α α β γ α β
Z[i]:
0 5 0 3 0 1
L = 7
R = 7
Z[i] = 1
Z[i]
i+Z[i]
S:
α β α β α β α α β γ α β
Z[i]:
0 5 0 3 0 1
L = 2
R = 6
Z[i] = 1
Z[i]
i+Z[i]
S:
α β α β α β α α β γ α β
Z[i]:
0 5 0 3 0 1
L = 2
R = 6
Z[i] = 2


Z[i]
i+Z[i]
S:
α β α β α β α α β γ α β
Z[i]:
0 5 0 3 0 1 2
L = 7
R = 8
Z[i] = 2
Z[i]
i+Z[i]
Z[i] = Z[i-L] = 0
S:
α β α β α β α α β γ α β
Z[i]:
0 5 0 3 0 1 2 0
L = 7
R = 8
Z[i] = 0
Z[i]
i+Z[i]
S:
α β α β α β α α β γ α β
Z[i]:
0 5 0 3 0 1 2 0 0
L = 7
R = 8
Z[i] = 0
Z[i]
i+Z[i]
S:
α β α β α β α α β γ α β
Z[i]:
0 5 0 3 0 1 2 0 0
L = 7
R = 8
Z[i] = 1
Z[i]
i+Z[i]
S:
α β α β α β α α β γ α β
Z[i]:
0 5 0 3 0 1 2 0 0 2
L = 10
R = 11
Z[i] = 2
Z[i]
Z[i] = Z[i-L] = 0
i+Z[i]
S:
α β α β α β α α β γ α β
Z[i]:
0 5 0 3 0 1 2 0 0 2 0
L = 10
R = 11
Z[i] = 0

Ερωτήσεις κατανόησης

Μπορούν δύο διαφορετικές συμβολοσειρές να έχουν τις ίδιες τιμές Z;
Ναι. Προφανώς άμα πάρουμε μία συμβολοσειρά και αλλάξουμε όλους τους χαρακτήρες ‘α’ σε ‘β’ και όλους τους χαρακτήρες ‘β’ σε ‘α’ τότε οι τιμές Z θα παραμείνουν οι ίδιες καθώς το μόνο που χρησιμοποιούμε στον αλγόριθμο είναι η ισότητα (και όχι διαφορετικού τύπου σύγκριση).

S:	α	β	α	β	β
Z[i]:	**5**	0	**2**	0	0

S:	β	α	β	α	α
Z[i]:	**5**	0	2	0	0

Για όλες τις συμβολοσειρές που έχουν ίδιες τιμές Z πρέπει να ισχύει ότι υπάρχει μετασχηματισμός 1-1 μεταξύ των χαρακτήρων της μίας και της άλλης[5]; Τι σημαίνει αυτό;

Η απάντηση είναι όχι. Δεν υπάρχει 1-1 μετασχηματισμός μεταξύ των παρακάτω παρόλο που έχουν τις ίδιες τιμές Z.

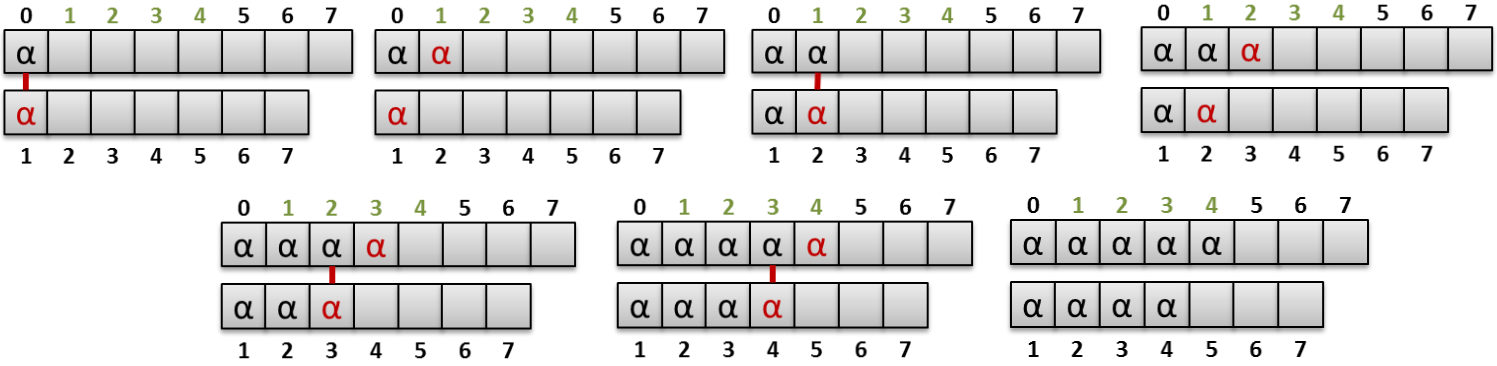
S:	α	β	β	β	β
Z[i]:	**5**	0	0	0	0

S:	α	β	γ	δ	ε
Z[i]:	**5**	0	0	0	0

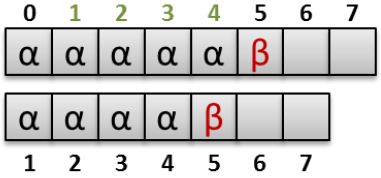
Αυτό σημαίνει ότι δεν μπορούμε να ανακτήσουμε μία συμβολοσειρά από τις τιμές Z.

Τι μπορούμε να πούμε για μία συμβολοσειρά που έχει $Z[1] = 4$;

Προφανώς, δεν ψάχνουμε να βρούμε συγκεκριμένους χαρακτήρες (δείτε την πρώτη ερώτηση). Επομένως χωρίς βλάβη της γενικότητας ας πούμε ότι ο πρώτος χαρακτήρας είναι ‘α’. Τότε από το $Z[1] = 4$ έχουμε ότι οι πρώτοι τέσσερεις χαρακτήρες κάνουν match τους χαρακτήρες από το 1..4. Η λογική που ακολουθούμε φαίνεται παρακάτω:



Επομένως κάθε φορά μπορούμε να υποστηρίξουμε ότι ο επόμενος χαρακτήρας είναι ‘α’ γιατί ξέρουμε το matching μεταξύ του prefix και του 1…4. Τελειώσαμε; Όχι, αν ο χαρακτήρας 5 ήταν ‘α’ τότε το $Z[1] = 5$, άρα πρέπει να τον θέσουμε κάτι διαφορετικό από ‘α’ (ή το τέλος της συμβολοσειράς).

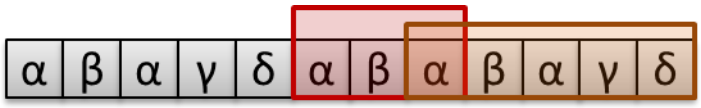


[5] Με απλά λόγια αν αντικαταστήσουμε κάθε χαρακτήρα της μίας με κάτι άλλο, χωρίς να μπορούν δύο χαρακτήρες να αντικατασταθούν από τον ίδιο.

Σημείωση: Εξίσου σημαντική με την έννοια του $Z[i]$ είναι το διάστημα $[i, i + Z[i] - 1]$. Αυτή είναι η υποσυμβολοσειρά που είναι ίση με κάποιο prefix της αρχικής και είναι το διάστημα που επηρεάζεται από την γνώση του $Z[i]$.

Μπορούμε να έχουμε δύο διαστήματα να τέμνονται (πριν δείτε την λύση βεβαιωθείτε ότι την έχετε προσπαθήσει αρκετά);

Ένα παράδειγμα μίας τέτοιας συμβολοσειράς είναι η ακόλουθη:



Η πιο απλή ιδιότητα των διαστημάτων μίας συμβολοσειράς S, είναι ότι $i + Z[i] < |S|$ καθώς θα πρέπει το διάστημα να τελειώνει πριν από το τέλος της S. Το ερώτημα που γεννάται είναι το ακόλουθο:

Τι σημαίνει για ένα prefix p να έχει $i + Z[i] < |p| - 1$ για όλα τα i;

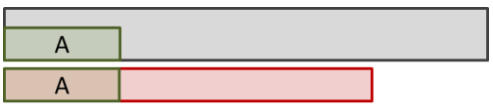
Σημαίνει ότι όλα τα διαστήματα τελειώνουν μέσα σε αυτή τη συμβολοσειρά, επομένως θα φαίνεται κάπως σαν το παρακάτω:

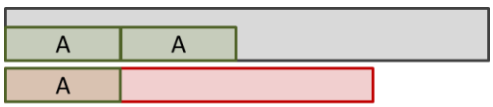
Για την υπόλοιπη συμβολοσειρά[6] έχουμε ότι όπου εμφανίζεται αυτό το prefix οι τιμές Z θα έχουν την ίδια μορφή. Ο λόγος είναι ο ίδιος με αυτή της περίπτωση 2.α στον αλγόριθμο, έχουμε ήδη βρει τον χαρακτήρα που σταματάει το διάστημα. Επομένως καμία τιμή Z δεν θα μπορέσει να ξεπεράσει το φράγμα του p. *Και πότε μπορεί;*

Μπορεί όταν υπάρχει μία τιμή στις Z τιμές του p που έχει $Z[i] = p - 1$. Ας μελετήσουμε αυτή την περίπτωση γιατί είναι μία με αρκετές εφαρμογές.

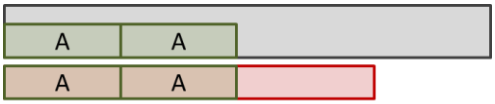
Σημαίνει ότι το p τελειώνει με ένα δικό του prefix. Αυτό είναι που του δίνει τη δυνατότητα να επεκτείνει τις Z τιμές του, όταν προσθέτουμε χαρακτήρες στο τέλος του. Ακόμα πιο ενδιαφέρον είναι άμα αρχίσουμε να κάνουμε match..

Ας ονομάσουμε Α το κομμάτι μεταξύ αρχής και του i για το οποίο $i + Z[i] = |S| - 1$.





[6] Καθώς το p είναι απλά ένα prefix



Επομένως, παρατηρούμε ότι υπάρχει μία περιοδικότητα. Η συνθήκη για το πότε βρήκαμε μία περίοδο της συμβολοσειράς είναι μέρος των προβλημάτων.