**Γιάννης Μπάρζας 2765**

**Ασκηση 1**

|  |  |  |
| --- | --- | --- |
|  | Χειροτερη | Αντισταθμιστικη |
| PUT | Ο(1)(σταθερο) | Ο(2) |
| GET | Ο(3)+Ο(1)+Ο(3m) | Ο(6) |
| MOVE | O(3m) | O(3m) |

Οπου **m** ποσα στοιχεια εχει η στοιβα S1.

Θεωρω ότι το κοστος isEmpty() εχει σταθερο κοστος Ο(1) αρα στην GET είναι Ο(3) καλειτε 3 φορες στα 2 if για κάθε εκτελεση της .

Στην MOVE η isEmpty() εχει κοστος Ο(**m** ) καθως ελενχετε **m** φορες στο while οσα και τα αντικειμενα μεσα στην S1.

Στην χειροτερη περιπτωση θεωρουμε τα κοστη για τα push ,pop ενός αντικειμενου Ο(1).

Στην **Αντισταθμιστικη αναλυση με Ενεργειακή μέθοδος για την PUSH:**

Το δυναμικο την στιγμη ι είναι Φ(Δι)

Αρχικα ουρες αδειες αρα Φ(Δ0)=0

s=1

Φ(Δι)= Φ(Δι-1)+1

S=s+ Φ(Δι)- Φ(Δι-1)=1+1=**2 Αντισταθμιστικο** κοστος για **PUSH**

Στην **Αντισταθμιστικη αναλυση με Ενεργειακή μέθοδος για την PΟP:**

Το δυναμικο την στιγμη ι είναι Φ(Δι)

Αρχικα ουρες αδειες αρα Φ(Δ0)=0

s=1

Φ(Δι)= Φ(Δι-1)-1

S=s+ Φ(Δι)- Φ(Δι-1)=1-1=**0** **Αντισταθμιστικο** κοστος για **PΟP**

Για την PUT εχουμε ένα **PUSH** οποτε στην αντισταθμιστικη εχει κοστος 2 (γιατι το **PUSH** εχει κοστος 2 για να αποπληρωσει και την **PΟP)**  ενώ στην χειροτερη κοστος 1 (σταθερο).

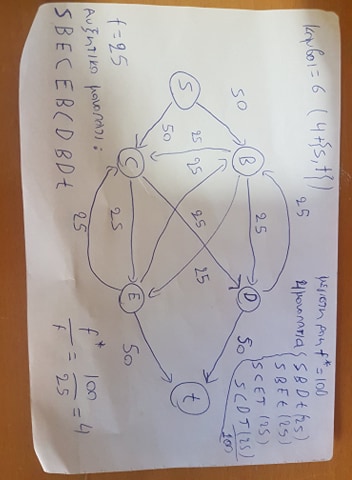
Η MOVE και στην αντισταθμιστικη και στην χειροτερη εχει κοστος O(3m) οπου **m** ποσα στοιχεια εχει η στοιβα S1 γιατι στην αντισταθμιστικη το **PUSH** εχει κοστος 2 ενώ το **PΟP 0** ενώ στην **χειροτερη** εχουν και τα δυο από κοστος 1 , συν το σταθερο κοστοςisEmpty().Για την **GET** στην χειροτερη περιπτωση ελενχουμε και τις 2 συνθηκες στα if καλειται η MOVE και κανει **PΟP** στο **return** συνολικο κοστος Ο(3)+Ο(1)+Ο(3m)

Ενώ στην αντισταθμιστικη η GET εχει κοστος m\* Ο(3)+Ο(3m)(απόMOVE) **για m εκτλεσεις** και το **PΟP** στο **return** εχει κοστος0**.**Αρα **για m εκτλεσεις της GET χειροτερη περιπτωσης** εχουμε **Ο(3m+3 m)** και αρα **Ο(3m+3 m)/m=O(6)** ανα μια εκτελεση GET .

O χρόνος εκτέλεσης χειρότερης περίπτωσης μιας αυθαίρετης ακολουθίας 𝑛 εντολών 𝑝𝑢𝑡 και 𝑔𝑒𝑡 είναι για ακολουθια n/2 𝑝𝑢𝑡 και n/2 𝑔𝑒𝑡 εναλλαξ εντολων είναι O(4n) ( n/2 put +n/2 get=>n/2 \* Ο(1)+ n/2 \* [ Ο(4)+Ο(3)]=Ο(n/2)+ν/2 \*[Ο(7)]=Ο(n/2+ 7 \* n/2)=Ο(8 \* n/2)=Ο(4n) .

**Ασκηση 2**

Αξιοποιωντας το παραδειγμα με την μεθοδο **Ford-Fulkerson** χωρις υπολοιπομενο δικτυο η ευρευση του |f| εξαρταται από την σειρα επιλογης του αυξητικου μονοπατιου.Με αλλα λογια αν επιλεχθει σαν πρωτο αυξητικο μονοπατι ένα μονοπατι που προκαλει ροη φραγης δηλαδη κορεσμο σε ενα συνολο από ακμες F στο οποιο υπαρχει τουλαχιστον μια ακμη από κάθε μονοπατι από τον s στον t ,τοτε δεν μπορουν να υπαρχουν αλλα αυξητικα μονοπατια και ο λογος της μεγιστη ροη f\* προς τη ροη που βρισκει f είναι μεγαλος συναρτηση των κομβων του δικτυου.Στο παραδειγμα της διαφανειας σε 27-30 υπαρχουν 4 κομβοι και ο λογος f\*/f είναι 2.Για ένα δικτυο με περισσοτερους κομβους αυτος ο λογος αυξανεται.Αυτο φαινεται και στο παρακατω παραδειγμα που το γραφημα εχει 6 κομβους και ο λογος είναι 4. Αυτό μας δειχνει ότι ο οσο αυξανονται οι κομβοι αυξανεται και ο λογος.



**Ασκηση 3**

**a)**

Αρχικα εχωντας ένα μη κατευθυνομενο γραφημα χωρις βαρη ή χωρητικοτητα στις ακμες βαζουμε σε ολες τις ακμες χωρητικοτητα ιση με 1 και βαζουμε για κάθε ακμη 2 αντιπαραλληλες με ιδια χωρητικοτητα με την αρχικη ετσι ώστε να εχουμε ένα κατευθυνομενο γραφημα.Στη συνεχεια εφαρμοζουμε τον αλγοριθμο **Ford-Fulkerson** στο κατευθυνομενο γραφημα που προκυπτει και βρισκουμε την μεγιστη ροη.Εχωντας αναθεσει χωρητικοτητα σε ολες τις ακμες ισο με 1 αρα οσο είναι η μεγιστη ροη είναι και τα ανεξαρτητα μονοπατια.Αρα ελενχουμε αν η μεγιστη ροη είναι μεγαλυτερη ή ιση του 2 και αν ναι τοτε εχουμε 2 ανεξαρτητα μονοπατια.Ο χρόνος εκτέλεσης είναι Ο(m f\*) όπου f\* η τιμή της μέγιστης ροής.

**b)**

Αρχικα εχωντας ένα μη κατευθυνομενο γραφημα χωρις βαρη -ή χωρητικοτητα στις ακμες βαζουμε σε ολες τις ακμες χωρητικοτητα ιση με 1 και βαζουμε για κάθε ακμη 2 αντιπαραλληλες με ιδια χωρητικοτητα με την αρχικη ετσι ώστε να εχουμε ένα κατευθυνομενο γραφημα.Στη συνεχεια εφαρμοζουμε τον αλγοριθμο **Edmonds-Karp** ο οποιος κρατα τα μονοπατιαστο κατευθυνομενο γραφημα που προκυπτει και βρισκουμε την μεγιστη ροη.Στο τελος της εκτελεσης επιστρεφουμε και τα 2 πρωτα ανεξαρτητα μονοπατια αν υπαρχουν.Εχωντας αναθεσει χωρητικοτητα σε ολες τις ακμες ισο με 1 αρα οσο είναι η μεγιστη ροη είναι και τα ανεξαρτητα μονοπατια.Αρα ελενχουμε αν η μεγιστη ροη είναι μεγαλυτερη ή ιση του 2 και αν ναι τοτε εχουμε 2 ανεξαρτητα μονοπατια.

Ο χρόνος εκτέλεσης είναι Ο((m^2) \*n) .

**Ασκηση 4**

Α)

Merge(A,B){

Result=[]

While(siza(A)!=0 and size(B)!=0){

If( min(A)>min(B) ){

X<-search( B,min(A) )

Z,B<-split(B,X) // Ζ στοχεια <=Χ

Join(Result,Z)

}else{

X<-search( A,min(B) )

Z,A<-split(A,X)

Join(Result,Z)

}

}(end while)

Ιf(size (A)==0){

Join(Result,B)

}else{

Join(Result,A)

}

Return Result;}

Θεωρω ότι η συνθηκη του While και του πρωτου If είναι σταθερη και ιση με Ο(2) και το δευτερο If , Return Ο(1). Search, Join, split ->O( logn)

Ο χρονος εκτελεσης χειροτερης περιπτωσης είναι

Ο(8n+6nlogn-2logn -2) δηλαδη περιπου **Ο(8n+6nlogn)** για δυο λιστες ιδιου μεγεθου με στοιχεια τετοια ώστε σε κάθε στιγμιοτυπο η λιστα με το ελαχιστο min από τις δυο λιστες εχει μονο ένα στοιχειο μικροτερο από το min της αλλης λιστας ετσι ώστε η λιστα Ζ στο ψευδοκωδικα να εχει κάθε φορα ένα στοιχειο μονο το μικροτερο min.Ετσι στην χειροτερη περιπτωση η επαναλυψη γινεται 2n-1 φορες αν κάθε φορα βγαζουμε ένα στοιχειο από κάθε λιστα εναλλαξ.

O( (2n-1)[4+3logn]+1+logn+1 )=

=O(6nlogn+8n-2logn-2) περιπου Ο(6nlogn+8n)

Β)

Φαρχικο=2nlogn

Σε κάθε συνχωνευση η διαφορα στο δυναμικο συνφωνα με την συναρτηση δυναμικου βλεπουμε ότι ένα ακριανο σημειο σε σχεση με ένα ενδιαμεσο εχει δυναμικο κατά logn παραπανω από ένα ενδιαμεσο Αυτό συνβαινει λογω του ότι στις 2 λιστες υπαρχουν 4 ακριανα στοιχεια αi πριν την συνχωνευση ενώ μετα την συνχωνευση υπαρχουν 2 ακριανα στοιχεια αi.

Σε μια οποιαδηποτε συνχωνευση 2 λιστων το δυναμικο πριν είναι

Φπριν=Φλιστας1+Φλιστας2 +Φ(υπολοιπων λιστων)

Το δυναμικο Φ(υπολοιπων λιστων) μενει σταθερο οποτε το διαφορα προκυπτει από τη διαφορα στο δυναμικο των λιστων που συνχωνευουμε.Πιο συγκεκριμενα :

Το δυναμικο των u, v είναι αυτό που μεταβαλεται επειδη μετα την συνχωνευση γινονται ενδιαμεσα στοιχεια από ακριανα.

Φ(u)= log(n)+log(aj+1-u) , Φ(v)= log(n)+log(v-aj-1)

Μετα την συνχωνευση:

Φ’(u)= log(u-x)+log(aj+1-u) , Φ’(v)= log(y-v)+log(v-aj-1)

Η διαφορα ΔΦ=Φ’-Φ=

= log(u-x)+log(aj+1-u) + log(y-v)+log(v-aj-1)-( log(n)+log(aj+1-u) + log(n)+log(v-aj-1) )=

= log(u-x)+ log(y-v)-2logn<=log( (x-y)/2 ) - 2logn ,επειδη η μεγαλυτερη διαφορα για τα u,v από τα χ ,y είναι (x-y)/2.

Σε 2 συνχωνευσεις η διαφορα στο δυναμικο είναι

ΔΦ<=2 log( (x-y)/2 ) - 4logn περιπου -1. #Αρα 2 συνχωνευσεις επιφερουν διαφορα δυναμικου -1.

Αντισταθμιστικα βαση της πολυπλοκοτητας της merge

και απο τα παραπανω σε μια εκτελεση της Merge γινονται περιπου O(logn) πραξεις αναζήτησης, διαχωρισμού και ένωσης ,αρα σε για m εκτελεσεις της Merge πραγματοποιουνται O( m logn) πραξεις αναζήτησης, διαχωρισμού και ένωσης και αρα το συνολικο κοστος για m εκτελεσεις της Merge είναι O( m logn) \* logn .Αρα για μια εκτελεση της Merge ο συνολικος χρονος είναι είναι O( m (log)^2(n) ) / m = O((log)^2(n) ).