סיכום למחשבים

פעולות בסיסיות לרשימות:

//מקבל רשימה ומספר ומוחק את ההופעה הראשונה של המספר ברשימה

public static Node<Integer> delFirstEncounter(Node<Integer> list,int num)

{

list = new Node<Integer>(-1,list);

Node<Integer> prv = list;

Node<Integer> p = list.getNext();

while (p!=null)

{

int x=p.getValue();

if(x==num)

{

prv.setNext(p.getNext());

list = list.getNext();

return list; //במידה ורוצים למחוק את כל ההופעות – למחוק את השורה הזאת

}

else

{

p=p.getNext();

prv = prv.getNext();

}

}

list = list.getNext();

return list;

//מקבל רשימה ומספר – מחזיר אמת אם המספר ברשימה אחרת שקר

public static boolean isIn(Node<Integer> list,int num)

{

Node<Integer>p = list;

while (p!=null)

{

int x=p.getValue();

if(x==num)

return true;

p=p.getNext();

}

return false;

}

//למחוק איבר מרשימה

p.setNext(p.getNext().getNext());

//מקבלת רשימה ומספר – מחזירה את הרשימה עם המספר בתחילת הרשימה

public static Node<Integer> addToBegin(Node<Integer> list,int x)

{

int y = list.getValue();

Node<Integer> p = new Node<Integer>(y,list.getNext());

list.setValue(x);

list.setNext(p);

return list;

}

//מקבל רשימה ומספר – מחזיר רשימה עם המספר בסופה

public static Node<Integer> addToEnd(Node<Integer> list, int x)

{

Node<Integer> p = list;

while(p.getNext()!=null)

p = p.getNext();

p.setNext(new Node<Integer>(x));

return list;

}

//בתחילת כל קוד

class Main {

public static void main(String[] args) {

public static void printList(Node<Integer>list)

{ //פעולה שמקבלת רשימה ומדפיסה אותה

Node<Integer>p=list;

while(p!=null)

{

int x=p.getValue();

System.out.print(x+" | ");

p=p.getNext();

}

System.out.println();

}

public static void buildList(Node<Integer>list,int []a)

{ // פעולה שמקבלת רשימה ריקה ומערך ובונה רשימה מאיברי המערך

Node<Integer>p=list;

for(int i=1; i<a.length; i++)

{

Node<Integer>x=new Node<Integer>(a[i]);

p.setNext(x);

p=p.getNext();

}

}

public static Node<Integer> getPrv(Node<Integer> list, Node<Integer> p)

{ //פעולה שמקבלת רשימה ומצביע לאיבר בה ומחזירה מצביע לאיבר שלפניו

Node<Integer> pList=list;

Node<Integer> next=pList.getNext();

while(next!=p && next!=null)

{

pList=next;

next=next.getNext();

}

return pList;

}

public static int sumList(Node<Integer> list)

{ // פעולה שמקבלת רשימה ומחזירה את הסכום שלה

int s=0;

Node<Integer>p=list;

while(p!=null)

{

int x=p.getValue();

s=s+x;

p=p.getNext();

}

return s;

}

public static void printTwoItemsList(Node<TwoItems>list)

{ // פעולה שתקבל רשימת זוגות ותדפיס אותה

Node<TwoItems>p=list;

while(p!=null)

{

TwoItems x=p.getValue();

System.out.println(x.getNumA()+" | "+x.getNumB());

p=p.getNext();

}

System.out.println();

}

public static int listLength(Node<Integer> list)

{ // פעולה שתקבל רשימה ותחזיר את מספר האיברים בה

Node<Integer> p=list;

int m=0;

while(p!=null)

{

m++;

p=p.getNext();

}

return m;

}

public static Node<Integer> getToPlace(Node<Integer> list, int x)

{ // פעולה שתקבל רשימה ומקום ותחזיר מצביע לאותו מקום ברשימה

Node<Integer> p=list;

int m=0;

while(p!=null && m!=x)

{

p=p.getNext();

m++;

}

return p;

}

פעולות בסיס בתורים:

//מקבל תור ומחזיר תור נוסף בלי להרוס את התור המקורי

public static Queue<Integer> makeCopy(Queue<Integer>q)

{

Queue<Integer>q1=new Queue<Integer>();

Queue<Integer>q2=new Queue<Integer>();

while(!q.isEmpty())

{

int x=q.remove();

q1.insert(x);

q2.insert(x);

}

while(!q1.isEmpty())

{

int x=q1.remove();

q.insert(x);

}

return q2;

}

פעולות פנימיות של רשימות:

List.getValue(); // לפי סוג הרשימה

list.getNext(); //Node

list.hasNext(); //boolean

list.setValue(value); //void

list.setNext(next); //void

פעולות פנימיות בתורים:

q.insert(x); //void

q.remove(); // לפי סוג הרשימה

q.top(); // לפי סוג הרשימה

q.isEmpty(); //boolean

נוסחת רנדום:

**int** f = (**int**) ((Math.random()\*(high-low+1))+low);

סטרינגים:

String str = "abc";

או

char data[] = {'a', 'b', 'c'};

String str = new String(data);

a = str.substring(i) //המשך הסטרינג מהקלט עד הסוף

Char - charAt(int index)

Returns the char value at the specified index.

מערך דו מימדי:

i =שורה mat.length;

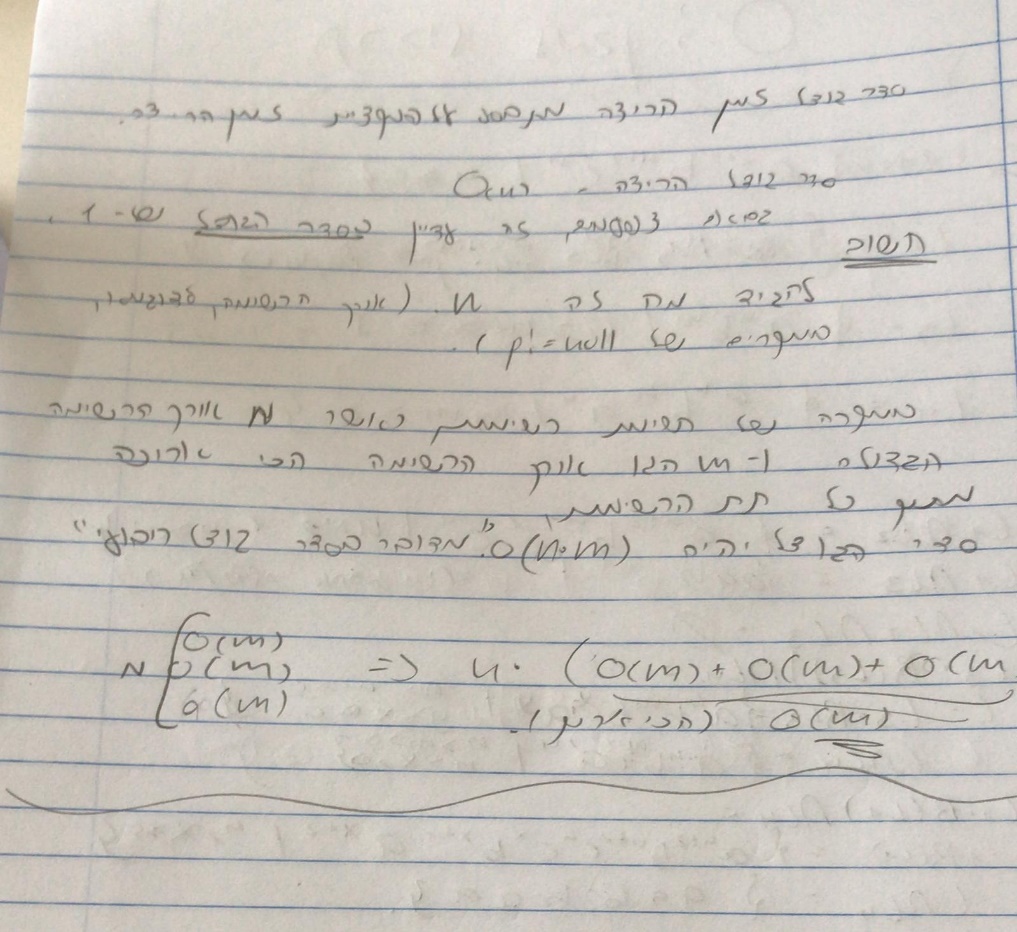
J = טור mat[].length;

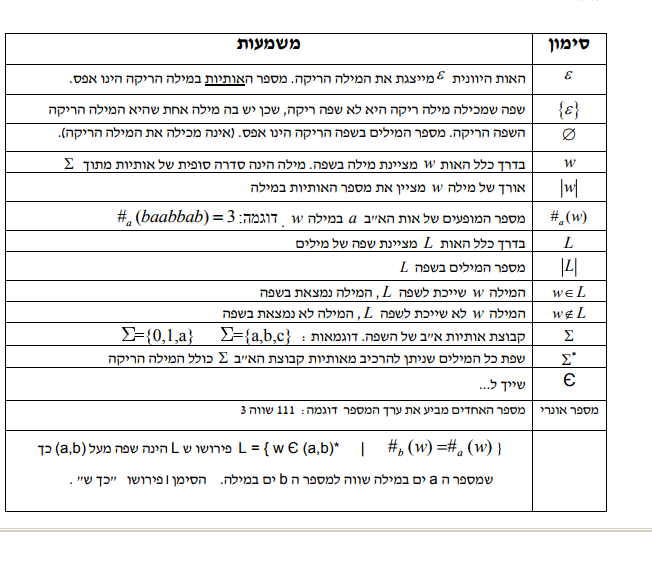
קלט:

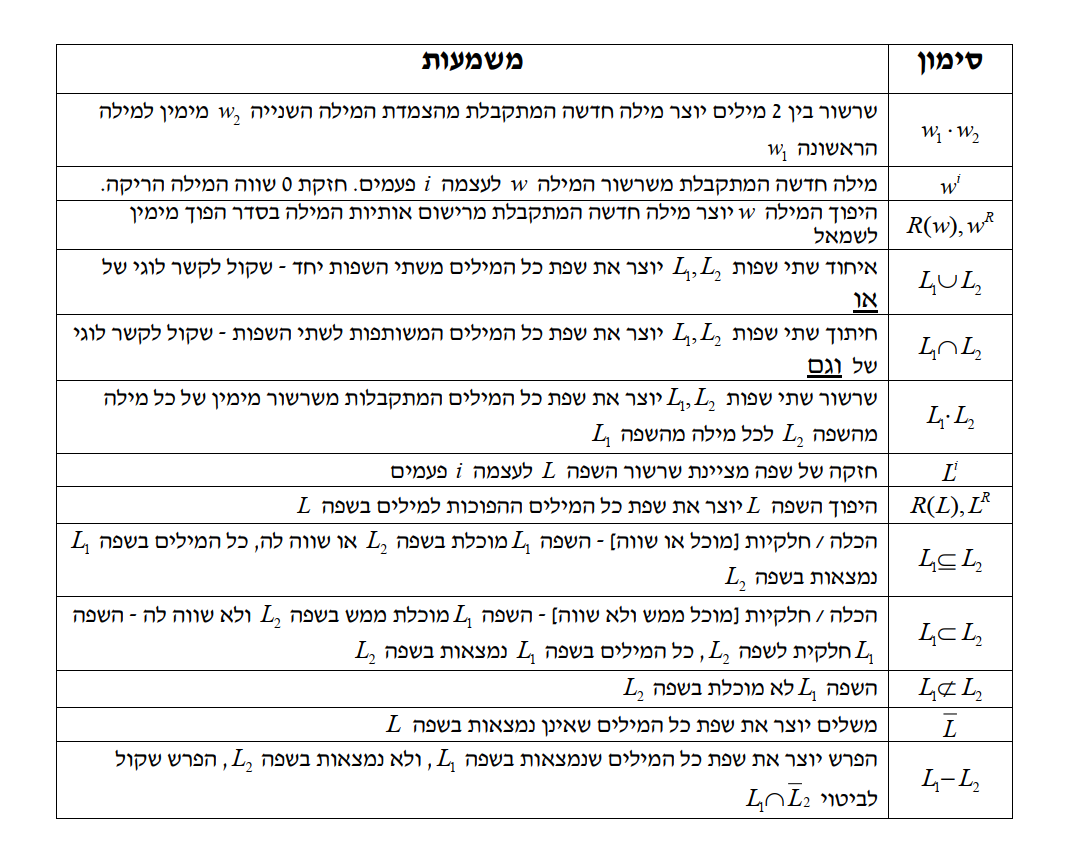
import java.util.\*;

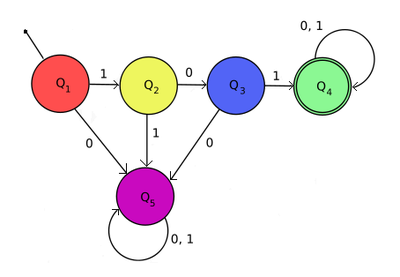
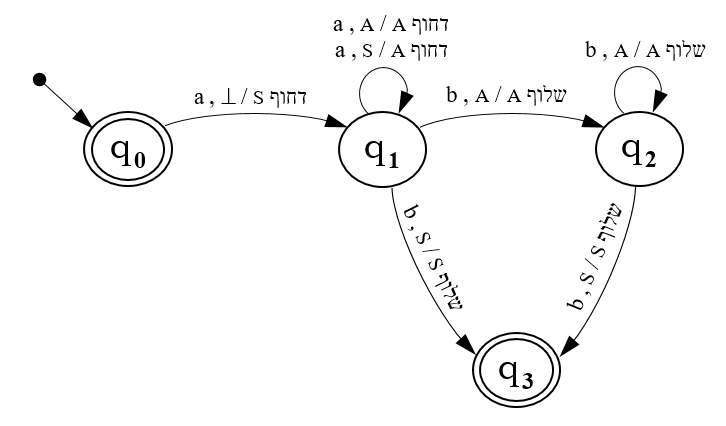
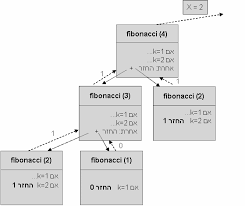
Scanner reader = new Scanner(System.in);

סיבוכיות:









טבלת מעקב לפעולה רקורסיבית המחזירה את המספר במיקום הn

אס"ד המקבל את כל המילים המתחילות ברצף 101

אוטומט מחסנית המקבל את כל המילים בעלות כמות a ולאחר מכן כמות b השווה לכמות הa