ANALIZA ALGORITMILOR

TEMA 1

NEGRU ADRIAN EDUARD 322CC

Problema 1.

1. Clase de complexitate

1.a $\quad n\boldsymbol{log^3n} = \text{o}\,(\boldsymbol{n^2})$

Calculam : $\lim\limits_{n\to\infty} \frac{\text{n}log^3n}{n^2}$. Simplificam si aplicam succesiv Teorema lui L'Hospital pe cazul $\frac{\infty}{\infty}$.

$\text{Lim}\limits_{n\to\infty} \frac{log^3n}{n} = \lim\limits_{n\to\infty} \frac{3log^2n}{nln2} = \frac{3}{ln2} \lim\limits_{n\to\infty} \frac{log^2n}{n} = \frac{3}{ln2} \lim\limits_{n\to\infty} \frac{2}{ln2}\ \frac{logn}{n} = \frac{3}{ln2}\ \frac{2}{ln2} \lim\limits_{n\to\infty} \frac{logn}{n} = \frac{3}{ln2}\ \frac{2}{ln2} \lim\limits_{n\to\infty} \frac{1}{\text{nln2}} = \frac{3}{ln2}\ \frac{2}{ln2}\ 0 = 0$

Deoarece $\lim\limits_{n\to\infty} \frac{\text{n}log^3n}{n^2} = 0 \ \rightarrow\ \text{n}log^3n = \text{o}\,(n^2)$.

1.b $\quad \boldsymbol{\log n! = \theta(nlogn)}$

$\exists\, c_1, c_2 \ \in R_+^*$, $\exists\, n_0 \ \in N^*$ a.i $\ c_1 nlogn \ \le logn! \ \le c_2\, nlogn$

i) Aleg $c_1 = \frac{1}{4}$.

⇨ $\frac{1}{4}nlogn \ \le logn!\,(1)$

$logn! = log1 + log2 + \ldots + logn$

Suma ultimei jumatati va fi mai mica decat intreaga suma

⇨ $log\frac{n}{2} + log(\frac{n}{2}+1) + \cdots + logn \le logn!$ (2)

Dar si suma din $log\frac{n}{2}$ de $\frac{n}{2}$ ori va fi mai mica decat relatia (2)

⇨ $\frac{n}{2}log\frac{n}{2} \le log\frac{n}{2} + log(\frac{n}{2}+1) + \cdots + logn$ (3)

Din (1), (2) si (3) :

⇨ $\frac{1}{4}nlogn \le \frac{n}{2}log\frac{n}{2} \le log\frac{n}{2} + log(\frac{n}{2}+1) + \cdots + logn \le logn!$

Deoarece relatiile (2) si (3) sunt adevarate pentru orice n $\in N^*$ $\mathit{prin\ tranzitivitate}$ ramane de aratat ca :

$\frac{1}{4}nlogn \le \frac{n}{2}log\frac{n}{2} \ \Leftrightarrow\ \frac{n}{2}log\sqrt{n} \le \frac{n}{2}log\frac{n}{2} \ \Leftrightarrow\ log\sqrt{n} \le log\frac{n}{2}$

Baza este 2 rezulta ca $\sqrt{n} \le \frac{n}{2}$ "A" pentru $\forall$ n $\ge$ 4 cu $n \in N^*$.

⇨ Pentru $c_1 = \frac{1}{4}$, $\exists\, n_0 \in N^*$, $n_0 = 4$ a.i (1) este "A"

ii) Aleg $c_2 = 1$

⇨ $logn! \leq nlogn$

$logn! = log1 + log2 + \ldots + logn \leq nlogn$ "A" pentru ∀ n ≥ 1 cu $n \in N^*$ deoarece majorez fiecare termen al sumei cu $logn$ si va rezulta o suma de $logn$ de n ori.

Conform cazurilor i) si ii) am demonstrat ca $\exists\, c_1 = \frac{1}{4}$, $c_2 = 1$ $c_1, c_2 \in R_+^*$, $\exists\, n_0 \in N^*$ $n_0 = 4$ astefel incat $c_1 nlogn \leq logn! \leq c_2\, nlogn$

⇨ $\log n! = \theta(nlogn)$

1.c $\mathbf{n! = \Omega(5^{\log n})}$

$\exists\, c_1 \in R_+^*$, $\exists\, n_0 \in N^*$ a.i $c_1 5^{\log n} \leq n!$

$5^{\log n} = n^{\log 5}$

$c_1 n^{\log 5} \leq c_1 n^{\log 8}$ "A" . Am majorat si am ales 8 pentru ca este putere a lui 2 .

Prin tranzitivitate trebuie sa demonstram : $c_1 n^{\log 8} \leq n!$ ⇔ $c_1 n^3 \leq n!$

Aleg c1 = $\frac{1}{5}$.

⇨ $\frac{1}{5}n^3 \leq n!$ ⇔

$n!$ = 1 ·2 · 3 · 4 · ... · (n-2) · (n-1) · n

⇔ $\frac{1}{5}n \cdot n \cdot n \leq$ 1 · 2 · 3 · 4 · ... · (n-2) · (n-1) · n

2 · 3 · 4 · (n-2) · (n-1) · n ≤ 1 · 2 · 3 · 4 · ... · (n-2) · (n-1) · n "A" ∀ n, cu $n \in N^*$

⇨ $\frac{1}{5}n \cdot n \cdot n \leq$ 2 · 3 · 4 · (n-2) · (n-1) · n ≤ 1 · 2 · 3 · 4 · ... · (n-2) · (n-1) · n .

Prin tranzitivitate ramane sa demonstram ca $\frac{1}{5}n \cdot n \cdot n \leq$ 2 · 3 · 4 · (n-2) · (n-1) · n .

$\frac{1}{5}n^2 \leq 24$ (n-2) (n-1) , iar $n \in N^*$

$\frac{\frac{1}{5}n^2}{24\ (n-2)(n-1)} \leq 1$, cu $n \in N^*$ $n \neq \{1,2\}$

⇨ Ramane de demonstrat ca $n^2 \leq 120\ (n-2)(n-1)$ ⇔ $0 \leq 119\, n^2 - 360n + 240$
Calculam derivata functiei si obtinem $238n - 360 = 0$.
Functia are punct de minim in n = $\frac{360}{238}$.

n	0	$\frac{360}{238}$	3 ∞
f ` (n)	- - - - - - - -	0	+ + + + + + + +
f(n)	↘	f($\frac{360}{238}$)	f(3) > 0 ↗

Deoarece functia este crescatoare cand $n \geq \frac{360}{238} \approx 1.51$. Aleg $n = 3$ si observ ca valoarea functiei este pozitiva, functia fiind crescatoare rezulta ca de la un rang $n_0 = 3$ relatia este adevarata pentru n natural.

⇨ $\exists\ c_1 = \frac{1}{5} \ \in R_+^* \ , \ \exists\ n_0 = 3 \ \in N^*$ a.i $c_1 5^{\log n} \leq n!$

⇨ $n! = \Omega\left(5^{\log n}\right)$

Problema 1.1 Algoritm

```
int function checkSum (a[ ], b[ ], x, sizeA, sizeB){
    if (a [sizeA - 1] <= b [ sizeB - 1] ){
        j ← sizeB – 1;
        i ← 0;
        d ← sizeA;
        caz ← 1;
    }
    else {
        j ← sizeA – 1;
        i ← 0;
        d ← sizeB;
        caz ← 2;
    }
    while (i < d  && j >= 0 ){
        if ( caz == 1)
            suma = a[i] + b[j]
        else if ( caz ==2)
            suma = a[j] + b[i]
        if (suma == x)
            return 1
```

```
            if (suma > x)
                    j = j - 1
            else
                    i = i + 1
      }
      return 0;
}
```

Complexitatea functiei este **O(n)** . Iterez de la stanga la dreapta si de la dreapta la stanga prin ambii vectori. . Decrementez pozitia din vectorul care contine numarul maxim si incrementez pozitia din celalalt vector. De fiecare data compar suma de pe cele doua pozitii cu numarul pe care vreau sa-l gasesc. Daca suma este mai mica incrementez valoarea lui i, iar daca suma este mai mare decrementez valoarea lui j. Algoritmul functioneaza pentru ca vectorii sunt sortati crescator. Deoarece parcurg cei 2 vectori de cel mult o data complexitatea este liniara **O(n).**

1.2 Gasirea si rezolvarea de recurente

- Pentru ambii algoritmi aleg 10^m $in\ O(1)$

a) Algoritm 1.

$$\mathbf{T(n)} = \begin{cases} \boldsymbol{k_1}, & \boldsymbol{k_1} \in R_+^*,\ \boldsymbol{n = 1} \\ \boldsymbol{4T\left(\frac{n}{2}\right) + k_2}, & \boldsymbol{k_2} \in R_+^*,\ \boldsymbol{n > 1} \end{cases}$$

Aplic metoda iterativa :

$$T(n) = 4\cancel{T\left(\frac{n}{2}\right)} + k_2 \quad |\ 4^0$$

$$\cancel{T\left(\frac{n}{2}\right)} = 4\cancel{T\left(\frac{n}{2^2}\right)} + k_2 \quad |\ 4^1$$

$$. \ . \ .$$

$$\cancel{T\left(\frac{n}{2^k}\right)} = 4T\left(\frac{n}{2^{k+1}}\right) + k_2 \quad |\ 4^k$$

(+)

$$\frac{n}{2^{k+1}} = 1 \quad \rightarrow \quad k = logn - 1$$

$$T(n) = 4^{k+1}\, T(1) + \ \theta(1)(\, 4^0 + 4^1 + \dots + 4^k) \qquad \Leftrightarrow$$

$$T(n) = 4^{k+1}\, T(1) + \frac{\theta(1)(\, 4^{k+1} - 1)}{3} \qquad \Leftrightarrow$$

$$T(n) = \theta\left(\, 4^{k+1} + \frac{(\, 4^{k+1} - 1)}{3}\right) \qquad \Leftrightarrow$$

$$T(n) = \theta\left(\frac{4^{k+2} - 1}{3}\right) \qquad \Leftrightarrow$$

$$T(n) = \theta\left(4^{k+2}\ \right) \qquad \Leftrightarrow$$

$$T(n) = \theta\left(4^{\log n + 1}\ \right) \qquad \Leftrightarrow$$

$$T(n) = \theta\left(n^{\log 4}\ \right) \qquad \Leftrightarrow$$

$$\boldsymbol{T(n) = \theta(n^2\,)}$$

b) Algoritm 2

$$\mathrm{T(n)} = \begin{cases} \boldsymbol{k_1}\,, & \boldsymbol{k_1} \in R_+^*\,,\ \boldsymbol{n = 1} \\ \boldsymbol{3T\left(\frac{n}{2}\right) + k_2}\,, & \boldsymbol{k_2} \in R_+^*\,,\ \boldsymbol{n > 1} \end{cases}$$

Aplic metoda iterativa :

$$T(n) = \ 3\cancel{T\left(\frac{n}{2}\right)} + \ k_2 \qquad |\ 3^0$$

$$\cancel{T\left(\frac{n}{2}\right)} = 3\cancel{T\left(\frac{n}{2^2}\right)} + \ k_2 \qquad |\ 3\text{^}1$$

$$. \ . \ .$$

$$\cancel{T\left(\frac{n}{2^k}\right)} = 3T\left(\frac{n}{2^{k+1}}\right) + k_2 \,|\ 3^k$$

(+)

$$\frac{n}{2^{k+1}} = 1 \quad \rightarrow \quad k = logn - 1$$

$$T(n) = 3^{k+1}\, T(1) + \theta(1)(3^0 + 3^1 + \dots + 3^k) \qquad \Leftrightarrow$$

$$T(n) = \theta\left(3^{k+1} + \frac{(3^{k+1}-1)}{2}\right) \qquad \Leftrightarrow$$

$$T(n) = \theta\left(\frac{3^{k+2} - 1}{2}\right) \qquad \Leftrightarrow$$

$$T(n) = \theta(3^{k+2}) \qquad \Leftrightarrow$$

$$T(n) = \theta(3^{\log n+1}) \qquad \Leftrightarrow$$

$$\boldsymbol{T(n) = \theta(n^{\log_2 3})}$$

1.3 Rezolvarea unei recurente

4. a) $\mathrm{T(n)} = \begin{cases} \boldsymbol{2a\,T(n-1), \quad n > 1, \; a} \in \boldsymbol{N^*} \\ \boldsymbol{2, \qquad\qquad\qquad n = 1} \end{cases}$

Aplic metoda iterativa :

$$T(n) = 2a\cancel{T(n-1)} \quad |\ (2a)^0$$

$$\cancel{T(n-1)} = 2a\cancel{T(n-2)}\ |\ (2a)^1$$

$$\dots$$

$$\cancel{T(2)} = 2aT(1) \quad |(2a)^{n-2}$$

(+)

$$T(n) = (2a)^{n-1}\, T(1) = \theta((2a)^{n-1})$$

$$\Rightarrow \boldsymbol{T(n) = \theta((2a)^{n-1})}$$

b) $T(n) = \begin{cases} \boldsymbol{3T\left(n^{\frac{1}{3}}\right) + logn}, & \boldsymbol{n > 1}, \\ \boldsymbol{1}, & \boldsymbol{n = 1} \end{cases}$

Facem o schimbare de variablia : $n = 2^m$

$T(2^m) = 3T\left(2^{\frac{m}{3}}\right) + log2^m$

$T(2^m) = X(m)$

$T\left(2^{\frac{m}{3}}\right) = X\left(\frac{m}{3}\right)$

Avem : $\boldsymbol{X(m) = 3X\left(\frac{m}{3}\right) + m}$ si aplicam T.Master.

Identificam:

a=3

b=3

f(m) = m

⇨ $m^{\log_b a} = m^{\log_2 2} = m$

Caz 2 Master :

⇨ f(m) = m $= \theta(m^{\log_b a}) = \theta(m) \rightarrow$ X(m) $= \theta(m^{\log_b a} lgm) = \theta(mlgm)$
⇨ $T(2^m) = X(m) = \theta(mlgm)$

Dar $n = 2^m$ ⇔ $m = lgn$

⇨ $\boldsymbol{T(n) = \theta(lgn\, lglgn)}$

5. $T(n) = \begin{cases} \boldsymbol{3T\left(\frac{n}{5}\right) + k_2 n^2}, & \boldsymbol{n > 1,\ k_2 \in R_+^*} \\ \boldsymbol{k_1}, & \boldsymbol{n = 1, k_1 \in R_+^*} \end{cases}$

Alpicam T. Master:

a=3

b=5 $\rightarrow$ $n^{\log_b a} = n^{\log_5 3}$ n

$f(n) = k_2 n^2$

Caz 2. $f(n) = k_2 n^2 = \theta\left(n^{\log_5 3}\right)$ "Fals"

Caz 1. $\exists\ \varepsilon > 0\ f(n) = O(n^{\log_b a-\varepsilon})$ atunci $T(n) = \theta(n^{\log_b a})$ (2)

$f(n) = O(n^{\log_5 3-\varepsilon})$ (1)

$f(n) = k_2 n^2$ (2)

$\log_5 3 - \varepsilon \le 1\ \forall\, \varepsilon > 0$

(2) Nu este posibil

Caz 3.

$\exists\ \varepsilon > 0\ a.i. f(n) = \Omega\left(n^{\log_b a+\varepsilon}\right),\ \exists\ c\ \in (0,1),\ \exists\ n_0 > 0, n_0\ \in\ N^*\ a.i\ af\left(\frac{n}{b}\right) \le cf(n)\ \ atunci\ T(n) = \theta(f(n))$

$Aleg\ \varepsilon = 1 > 0$.

Demonstram ca relatia $f(n) = \Omega\left(n^{\log_b a+\varepsilon}\right)$ $este\ adevarata$:

$\exists\, c_1\ \in R_+^*, \exists\, n_0\ \in N^*\ a.i\ c_1\, n^{\log_5 3+1} \le f(n)$ ⇔

$c_1\, n^{\log_5 3+1} \le k_2 n^2$

Aleg $c_1 = k_2\ \in R_+^*$ (1)

⇨ $n^{\log_5 3+1} \le n^2$

⇨ $n^{\log_5 3-1} \le 1$ "A" $\forall\, n_0\ \in\ N^*$ (2)

⇨ $Din\ (1)\ si\ (2)\ ca . f(n) = \Omega\left(n^{\log_b a+\varepsilon}\right),$

$af\left(\frac{n}{b}\right) \le cf(n)$ ⇔ $3f\left(\frac{n}{5}\right) \le cf(n)$ ⇔ $3\, k_2 (\frac{n}{5})^2 \le c\, k_2 n^2$ ⇔ $\frac{3}{25} \le c,\ \ c\ \in (0,1)$

Aleg c = $\frac{3}{25}$

⇨ $\exists\ \varepsilon = 1 > 0\ a.i. f(n) = \Omega\left(n^{\log_b a+\varepsilon}\right),\ \exists\ c = \frac{3}{25}\ \in (0,1),\ \exists\ n_0 > 0, n_0\ \in N^*\ a.i\ af\left(\frac{n}{b}\right) \le cf(n)$

⇨ $\boldsymbol{T(n) = \theta(n^2)}$

Metoda Substitutiei .

Aleg $\boldsymbol{T(n) = \theta(n^2)}$

$T(n) = 3T\left(\frac{n}{5}\right) + k_2 n^2, \qquad T(1) = k_1$

$\exists\, c_1, c_2 \in R_+^*$, $\exists\, n_0 \in N^*$ a.i $c_1 n^2 \le T(n) \le c_2\, n^2$

Caz de baza :

n = 1 $\quad c_1 \le T(1) \le c_2 \quad n_0 = 1 \quad "A"$

Pas de inductie : $\boldsymbol{\frac{n}{5} \to n}$

Ipoteza de inductie: $c_1 (\frac{n}{5})^2 \le T(\frac{n}{5}) \le c_2\, (\frac{n}{5})^2$

Arat ca : $c_1 n^2 \le T(n) \le c_2\, n^2$

$c_1 (\frac{n}{5})^2 \le T(\frac{n}{5}) \le c_2\, (\frac{n}{5})^2 \qquad | \cdot 3$

$3c_1 (\frac{n}{5})^2 \le 3T(\frac{n}{5}) \le 3c_2\, (\frac{n}{5})^2 \quad | + k_2 n^2$

$3c_1 (\frac{n}{5})^2 + k_2 n^2 \le T(n) \le 3c_2 \left(\frac{n}{5}\right)^2 + k_2 n^2 \quad \Leftrightarrow$

$c_1 n^2 - \frac{22}{25} c_1 n^2 + k_2 n^2 \le T(n) \le c_2\, n^2 - \frac{22}{25} c_2\, n^2 + k_2 n^2$

⇨ $\begin{cases} -\frac{22}{25} c_1 n^2 + k_2 n^2 > 0 \\ -\frac{22}{25} c_2 n^2 + k_2 n^2 < 0 \end{cases} \Leftrightarrow \begin{cases} k_2 > \frac{22}{25} c_1 \\ k_2 < \frac{22}{25} c_2 \end{cases} \Leftrightarrow \frac{22}{25} c_1 < k_2 < \frac{22}{25} c_2$

⇨ $c_1 < \frac{25}{25} k_2 < c_2$

⇨ $(\exists) \begin{cases} c_1 = \min(k_1, \frac{25}{25} k_2) \\ c2 = \max(k_1, \frac{25}{25} k_2) \end{cases} \in R_+^*,\ (\exists)\, n_0 \in N^*\ astfel\ incat \quad c_1 n^2 \le T(n) \le c_2\, n^2$ “A”

⇨ Din inductie rezulta ca $\boldsymbol{T(n) = \theta(n^2)}$

Problema 2. Algoritmul lui Dijkstra.

Cum am numarat operatiile elementare ?

k++ ;	Atribuirea din for (i=0) → 1
for(i=0; i<n; i++){	O comparatie si o incrementare → 3
}	Inainte de a iesi din for mai face o comparatie → 1
Structura. camp	O operatie elementara (accesare) → 1
v[i + 1]	O adunare si o accesare → 2
v[g.n] = i / 2	Doua accesari o asignare si o impartire → 4

Nu am luat ca fiind operatie elementara dereferentierea.

	Matrice vector		**Liste vector**		**Matrice heap**		**Liste heap**	
Test	*Operatii*	*Timp*	*Operatii*	*Timp*	*Operatii*	*Timp*	*Operatii*	*Timp*
Test01	2863	0m0.001s	1649	0m0.001s	2143	0m0.001s	1061	0m0.001s
Test02	4185	0m0.001s	2292	0m0.001s	3032	0m0.001s	1414	0m0.001s
Test03	4546	0m0.001s	3023	0m0.001s	3570	0m0.001s	2331	0m0.001s
Test04	30434	0m0.001s	20000	0m0.001s	20832	0m0.001s	11224	0m0.001s
Test05	879609	0m0.003s	743680	0m0.004s	549307	0m0.002s	500230	0m0.004s
Test06	197184	0m0.001s	165633	0m0.001s	131381	0m0.001s	114021	0m0.001s
Test07	25114224	0m0.010s	3023726	0m0.008s	23048147	0m0.009s	1221572	0m0.007s
Test08	23245641	0m0.008s	394757	0m0.007s	23039943	0m0.008s	210138	0m0.007s
Test09	13216788	0m0.005s	384445	0m0.003s	12968400	0m0.005s	157914	0m0.003s

Se observa ca numarul de operatii este mai mare in cazul in care se foloseste o matrice pentru a reprezenta graful. Implementarea cu liste este mai optima din punct de vedere al timpului si al numarului de operatii efectuate. Diferenta este si mai bine evidentiata in testele 07,08,09 atunci cand graful este rar. Fiind rar, in matrice vor fi multe zerouri, iar matricea va fi parcursa secvential, multe zerouri ducand la multe verificari care oricum nu vor fi luate in considerare. Spre deosebire de matrice, atunci cand folosim liste introducem doar vecinii si vom parcurge doar nodurile care ne intereseaza. Diferenta de eficienta este mai mica in cazul testelor 04, 05, 06 deoarece graful este dens si matricea nu contine multe zerouri.

Valorile apropiate sunt in cazul in care graful este dens. Vecinii unui nod sunt accesati la matrice direct a[i][j], iar la liste vecinii sunt accesati prin nod→next (fiind liste de adiacenta se introduc numai vecinii unui nod).

Din testele efectuate cea mai optima implementare este cea cu liste si min heap. Eficienta listelor este mentionata mai sus, iar eficienta heapurilor este data de timpul de accesare al minimului acesta realizandu-se in O(1) .

	Var.	Cost	Drum
Test 01	1	5	0 1 4 6 5
	2	5	0 1 4 6 5
	3	5	0 1 4 6 5
	4	5	0 1 4 6 5
Test 02	1	29	0 8 7 2 1 9 5
	2	29	0 8 7 2 1 9 5
	3	29	0 8 7 2 1 9 5
	4	29	0 8 7 2 1 9 5
Test03	1	15	0 6 1 5
	2	15	0 6 1 5
	3	15	0 6 1 5
	4	15	0 6 1 5
Test 04	1	12	12 13 20 1
	2	12	12 13 20 1
	3	12	12 13 20 1
	4	12	12 13 20 1
Test 05	1	569	47 0 7 19
	2	569	47 0 7 19
	3	569	47 0 7 19
	4	569	47 0 7 19
Test 06	1	7	34 32 18 39
	2	7	34 32 18 39
	3	7	34 32 18 39
	4	7	34 32 18 39
Test 07	1	49524	310 791 881 5 14 177 546 994 267 354 690 394 8 563 548 248 256 974 402 221 90 752 314 53 65 97 624 738 154 675 230 611 812 523 423 997 239 556 979 280 904 632 75 643 460 186 933 328 442 598 422 647 646 705 975 213 802 458 465 560 91

			244 786 0 852 907 656 649 122 333 807 111 136 712 364 9
	2	49524	310 791 881 5 14 177 546 994 267 354 690 394 8 563 548 248 256 974 402 221 90 752 314 53 65 97 624 738 154 675 230 611 812 523 423 997 239 556 979 280 904 632 75 643 460 186 933 328 442 598 422 647 646 705 975 213 802 458 465 560 91 244 786 0 852 907 656 649 122 333 807 111 136 712 364 9
	3	49524	310 791 881 5 14 177 546 994 267 354 690 394 8 563 548 248 256 974 402 221 90 752 314 53 65 97 624 738 154 675 230 611 812 523 423 997 239 556 979 280 904 632 75 643 460 186 933 328 442 598 422 647 646 705 975 213 802 458 465 560 91 244 786 0 852 907 656 649 122 333 807 111 136 712 364 9
	4	49524	310 791 881 5 14 177 546 994 267 354 690 394 8 563 548 248 256 974 402 221 90 752 314 53 65 97 624 738 154 675 230 611 812 523 423 997 239 556 979 280 904 632 75 643 460 186 933 328 442 598 422 647 646 705 975 213 802 458 465 560 91 244 786 0 852 907 656 649 122 333 807 111 136 712 364 9
Test 08	1	27	120 130 501 500 208
	2	27	120 130 501 500 208
	3	27	120 130 501 500 208
	4	27	120 130 501 500 208
Test 09	1	1299	709 508 113 136 220 78
	2	1299	709 508 113 136 220 78
	3	1299	709 508 113 136 220 78
	4	1299	709 508 113 136 220 78