

Benchmark

Javolution vs. Trove vs. Java

Compared implementations:

Javolution: javolution-4.2.6.jar { FastList | FastMap }

Trove: trove-2.0.2.jar { TIntArrayList | TIntObjectHashMap }

Java: build 1.6.0_03-b05 { ArrayList | HashMap }

Average values given in 1000 units

ListAdd

Javolution1: int -> FastList<Integer>

Javolution2: Integer -> FastList<Integer>

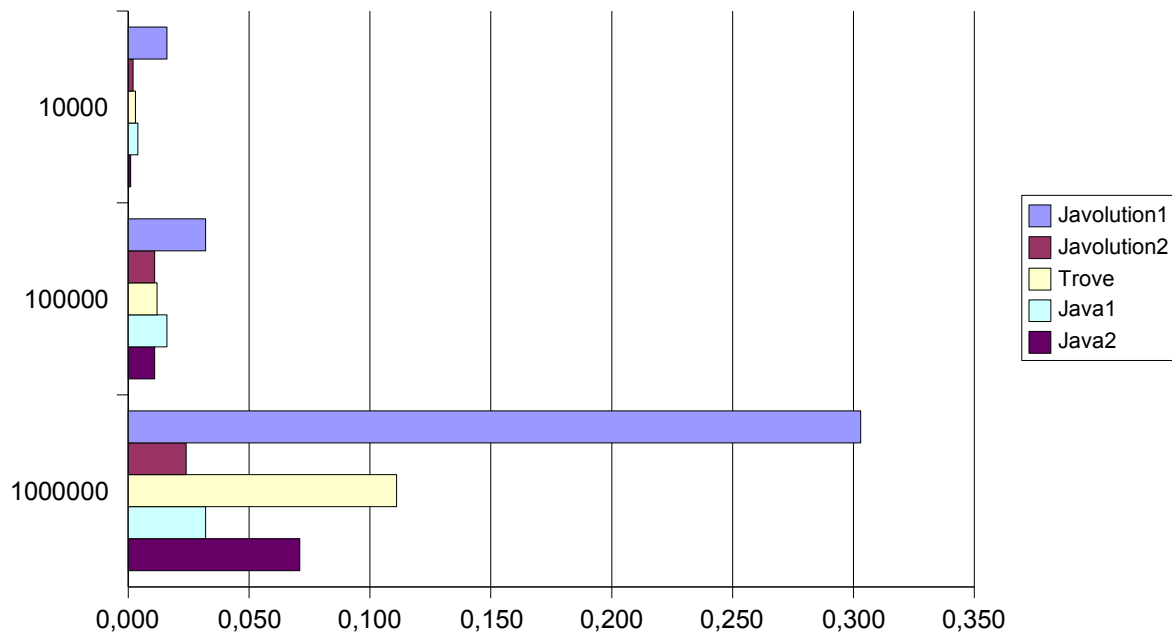
Trove: int -> TIntArrayList

Java1: int -> ArrayList<Integer>

Java2: Integer -> ArrayList<Integer>

Runs	Javolution1	Javolution2	Trove	Java1	Java2
10000	0,016	0,002	0,003	0,004	0,001
100000	0,032	0,011	0,012	0,016	0,011
1000000	0,303	0,024	0,111	0,032	0,071
Average	0,0001	0,0000	0,0000	0,0000	0,0000

ListAdd

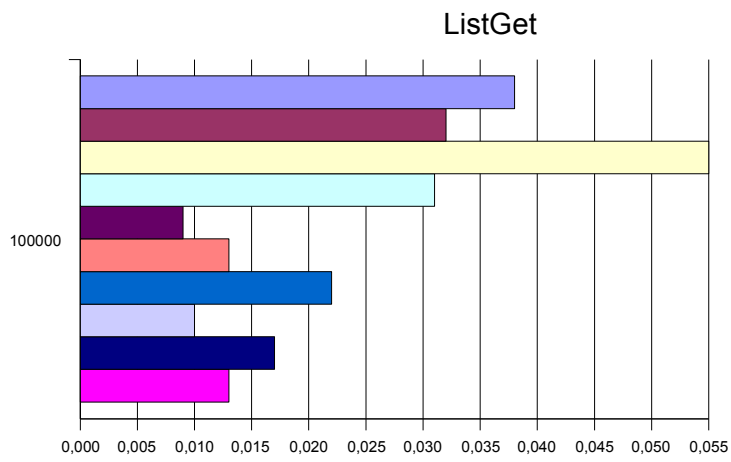
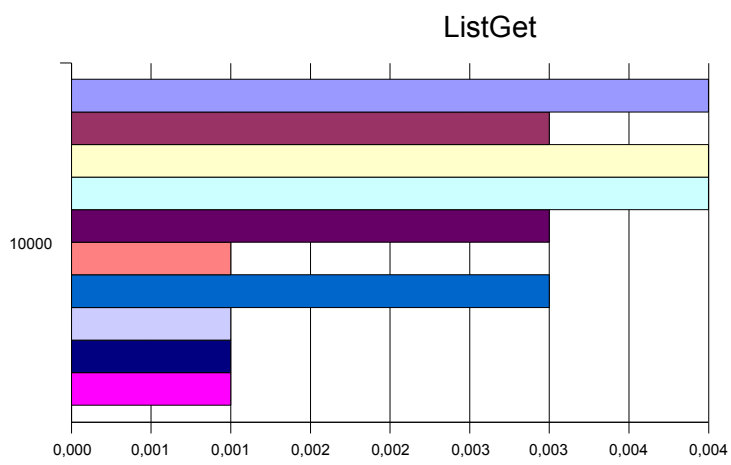
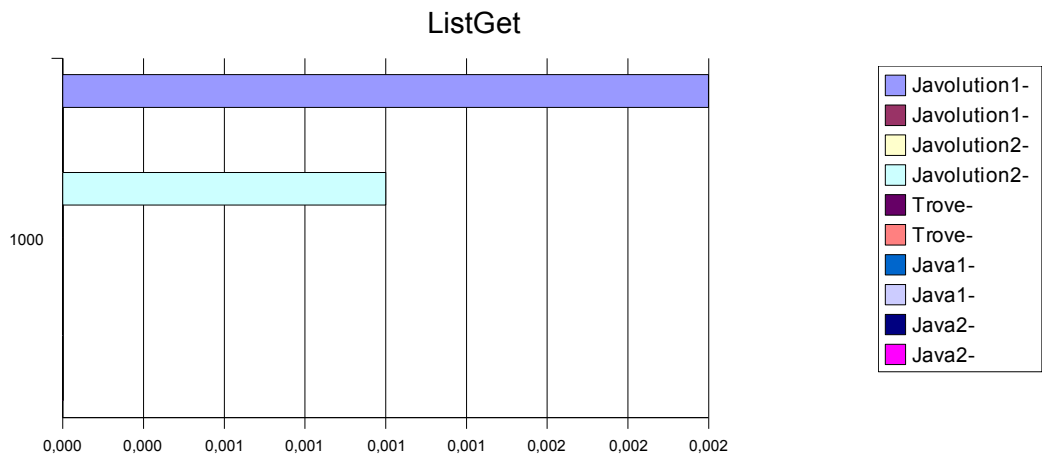


ListGet

Javolution1-1: int <- FastList(int)
Javolution1-2: Integer <- FastList(int)
Javolution2-1: int <- FastList(Integer)
Javolution2-2: Integer <- FastList(Integer)
Trove-1: int <- TIntArrayList(int)
Trove-2: Integer <- TIntArrayList(int)
Java1-1: int <- ArrayList(int)
Java1-2: Integer <- ArrayList(int)
Java2-1: int <- ArrayList(Integer)
Java2-2: Integer <- ArrayList(Integer)

Runs	Javolution1-1	Javolution1-2	Javolution2-1	Javolution2-2
1000	0,002	0,000	0,000	0,001
10000	0,004	0,003	0,004	0,004
100000	0,038	0,032	0,055	0,031
Average	0,0001	0,0000	0,0000	0,0001

Runs	Trove-1	Trove-2	Java1-1	Java1-2	Java2-1	Java2-2
1000	0,000	0,000	0,000	0,000	0,000	0,000
10000	0,003	0,001	0,003	0,001	0,001	0,001
100000	0,009	0,013	0,022	0,010	0,017	0,013
Average	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000



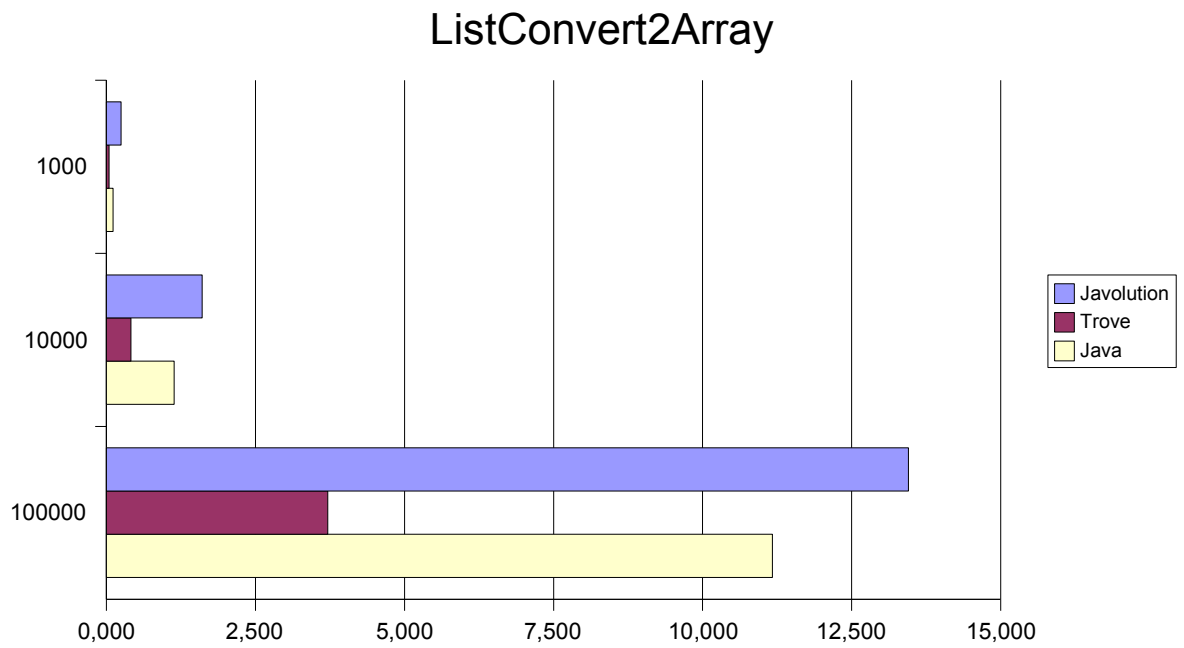
ListConvert2Array

Javolution: FastList<Integer> -> Integer[]

Trove: TIntArrayList -> int[]

Java: ArrayList<Integer> -> Integer[]

Runs	Javolution	Trove	Java
1000	0,002	0,000	0,000
10000	0,021	0,006	0,006
100000	0,001	0,010	0,014
Average	0,0180	0,0040	0,0112



ListContains

Javolution1: FastList(int)

Javolution2: FastList(Integer)

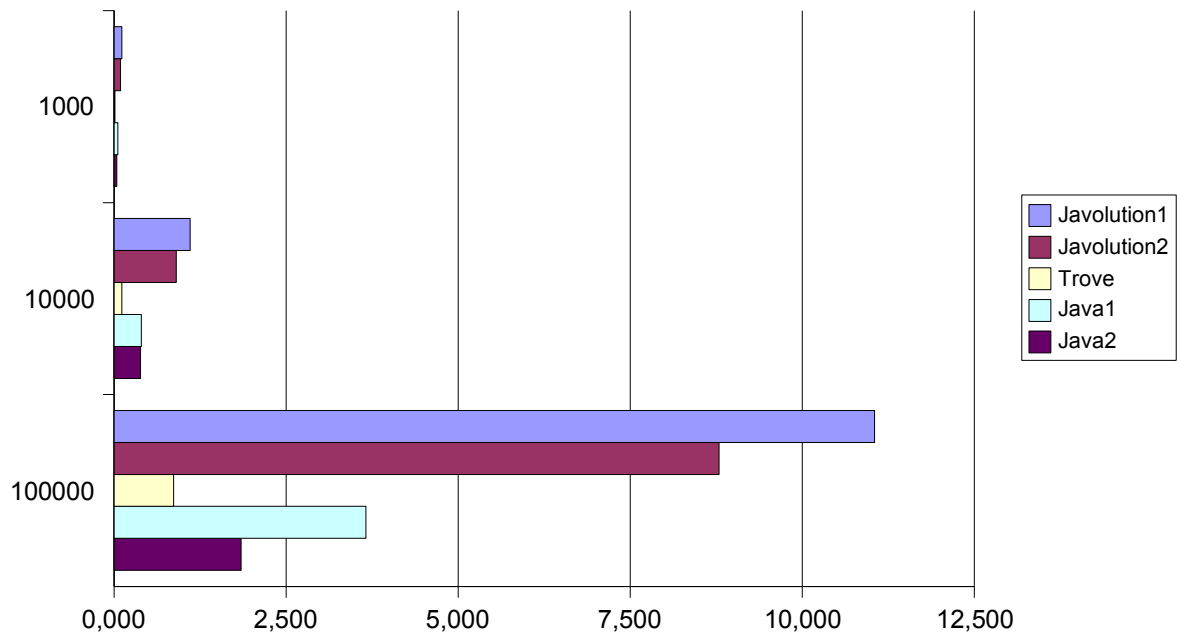
Trove: TIntArrayList(int)

Java1: ArrayList(int)

Java2: ArrayList(Integer)

Runs	Javolution1	Javolution2	Trove	Java1	Java2
1000	0,112	0,094	0,014	0,053	0,039
10000	1,105	0,902	0,112	0,395	0,385
100000	11,049	8,789	0,866	3,658	1,844
Average	0,0111	0,0091	0,0011	0,0043	0,0032

ListContains



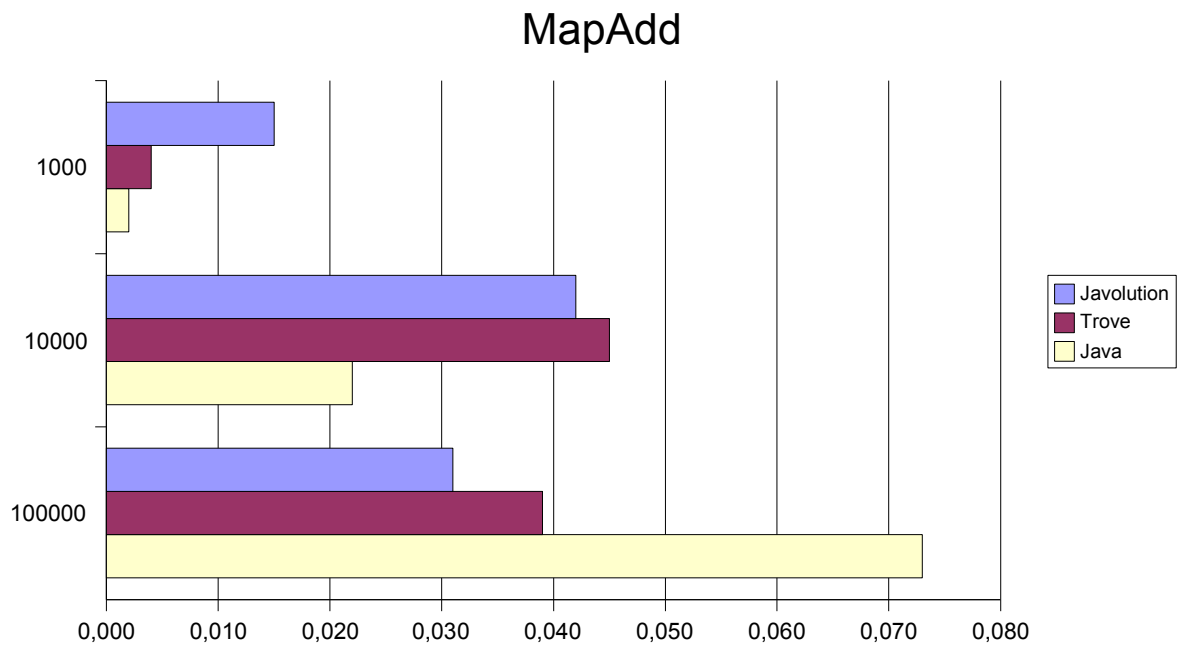
MapAdd

Javolution: FastMap<Integer, L2Polygon>

Trove: TIntObjectHashMap<int, L2Polygon>

Java: HashMap<Integer, L2Polygon>

Runs	Javolution	Trove	Java
1000	0,015	0,004	0,002
10000	0,042	0,045	0,022
100000	0,031	0,039	0,073
Average	0,0007	0,0003	0,0002



MapGet

Javolution1: FastMap(int)

Javolution2: FastMap(Integer)

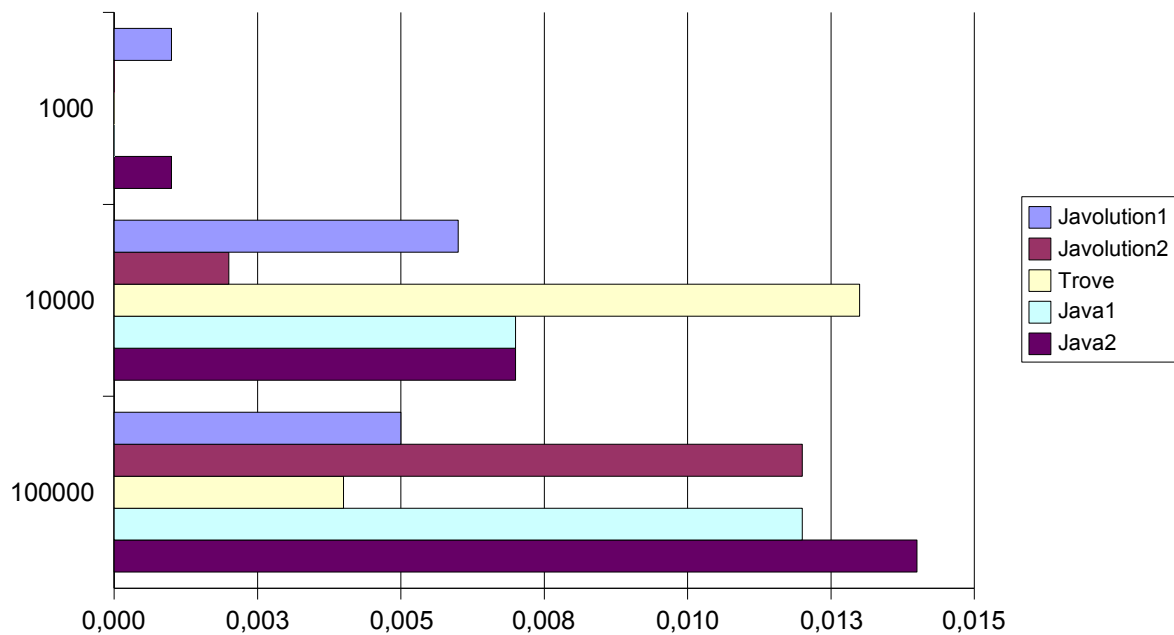
Trove: TIntObjectHashMap(int)

Java1: HashMap(int)

Java2: HashMap(Integer)

Runs	Javolution1	Javolution2	Trove	Java1	Java2
1000	0,001	0,000	0,000	0,000	0,001
10000	0,006	0,002	0,013	0,007	0,007
100000	0,005	0,012	0,004	0,012	0,014
Average	0,0001	0,0000	0,0000	0,0000	0,0001

MapGet



MapConvert

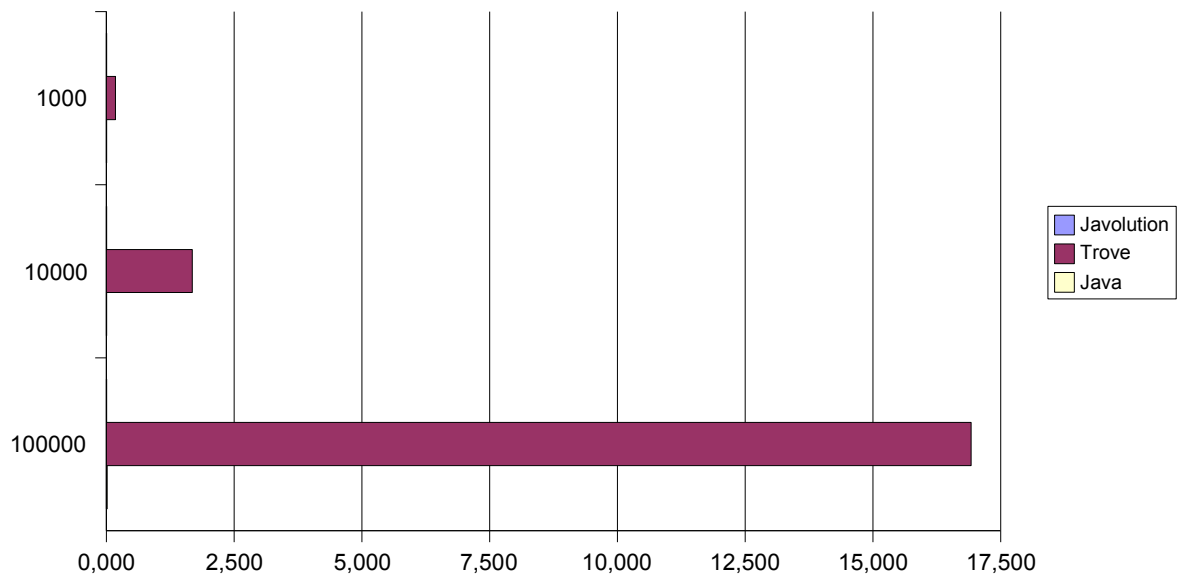
Javolution: Collection <- FastMap()

Trove: Object[] <- TIntObjectHashMap()

Java: Collection <- HashMap()

Runs	Javolution	Trove	Java
1000	0,002	0,179	0,000
10000	0,002	1,679	0,002
100000	0,005	16,922	0,013
Average	0,0001	0,0172	0,0000

MapConvert



ContainsKey

Javolution1: FastMap(int)

Javolution2: FastMap(Integer)

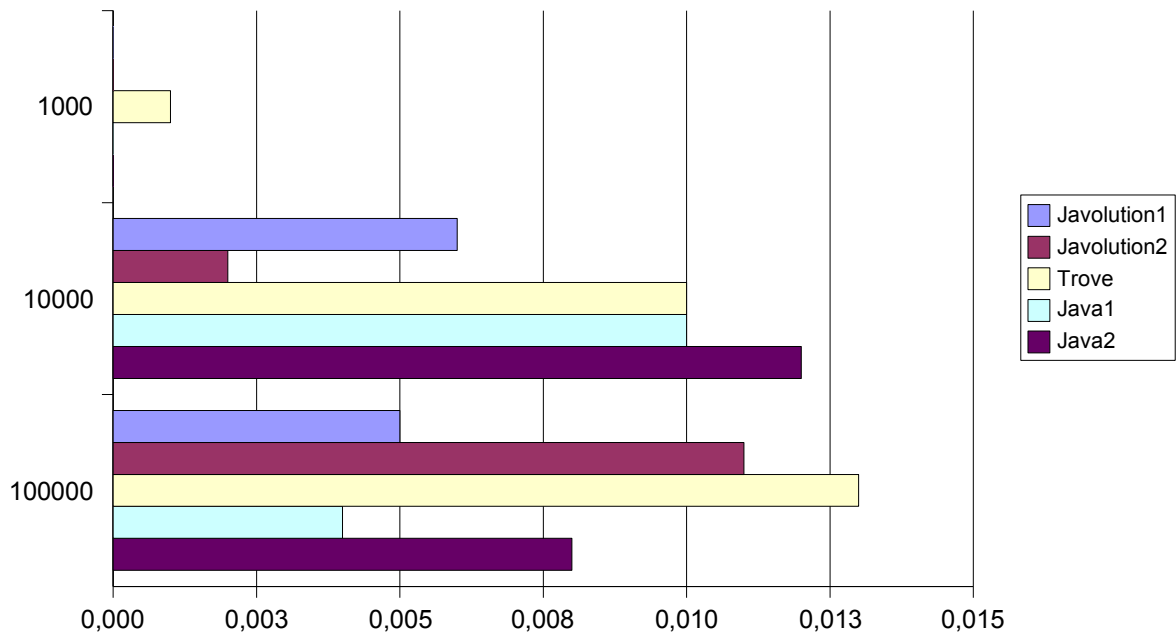
Trove: TIntObjectHashMap(int)

Java1: HashMap(int)

Java2: HashMap(Integer)

Runs	Javolution1	Javolution2	Trove	Java1	Java2
1000	0,000	0,000	0,001	0,000	0,000
10000	0,006	0,002	0,010	0,010	0,012
100000	0,005	0,011	0,013	0,004	0,008
Average	0,0000	0,0000	0,0001	0,0000	0,0000

MapContainsKey



ContainsValue

Javolution: FastMap(L2Polygon)

Trove: TIntObjectHashMap(L2Polygon)

Java: HashMap(L2Polygon)

Runs	Javolution	Trove	Java
1000	0,002	0,000	0,000
10000	0,021	0,006	0,006
100000	0,001	0,010	0,014
Average	0,0001	0,0000	0,0000

MapContainsValue

