Question 3:
The code for the problem is present in the attached github repository. We could note that we have 3 nested for loops and we can interchange them maintaining the precision of the solution. Following are the combinations of the loop and their performance

Note: *Time in Nanoseconds.*

I, J, K

N	128	256	512
Iteration 1	11238176	87735141	738401854
Iteration 2	10315913	95228120	724739624
Iteration 3	10026547	88626850	716162352
Iteration 4	10444267	86697339	712669493
Iteration 5	10741363	114325474	734300589
Median	**10444267**	**88626850**	**724739624**

I, K, J

N	128	256	512
Iteration 1	9573563	59786247	420068011
Iteration 2	9467350	59430480	425174818
Iteration 3	11856721	58671657	419498810
Iteration 4	9306361	58621853	386511881
Iteration 5	9381577	59718417	420470088
Median	**9467350**	**59430480**	**420068011**

K, I, J

N	128	256	512
Iteration 1	9981769	58333633	422652882
Iteration 2	9305010	61117587	431034842

Iteration 3	8876656	59704946	426423398
Iteration 4	10132701	59349419	423859587
Iteration 5	8090361	62369342	422590950
Median	**9305010**	**59704946**	**423859587**

K, J, I

N	128	256	512
Iteration 1	13794999	102065793	814084227
Iteration 2	9774119	97619202	798926941
Iteration 3	13413033	103121638	852852350
Iteration 4	12716718	98763877	817270906
Iteration 5	10712025	108070460	811043671
Median	**12716718**	**102065793**	**814084227**

J, K, I

N	128	256	512
Iteration 1	12023385	103437377	812519930
Iteration 2	11741502	107421633	823129128
Iteration 3	13631846	101075464	828516363
Iteration 4	13316720	100282157	817735952
Iteration 5	12827583	103077558	797951845
Median	**12827583**	**103077558**	**817735952**

J, I, K

N	128	256	512
Iteration 1	9882507	89970305	750988468
Iteration 2	10625240	87557448	728962395

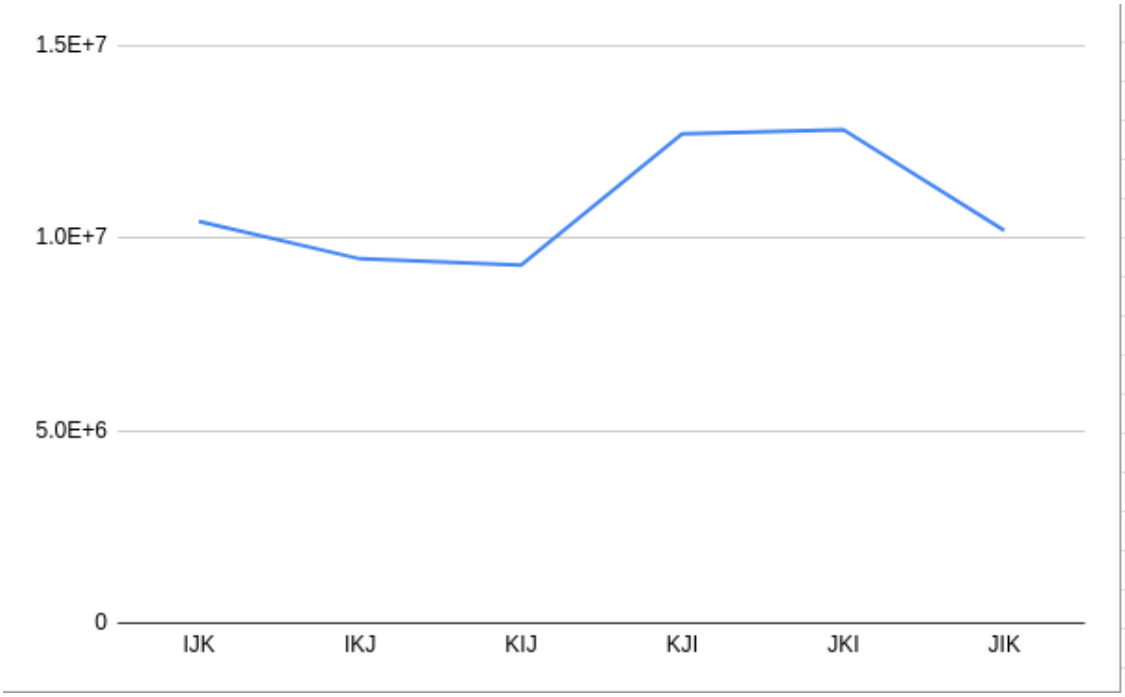
Iteration 3	10200303	90462679	743842431
Iteration 4	9880655	92903623	782598922
Iteration 5	12014748	87495620	724580132
Median	**10200303**	**89970305**	**743842431**

SUMMARY:
The summary of the above experiment is shown in the following table:

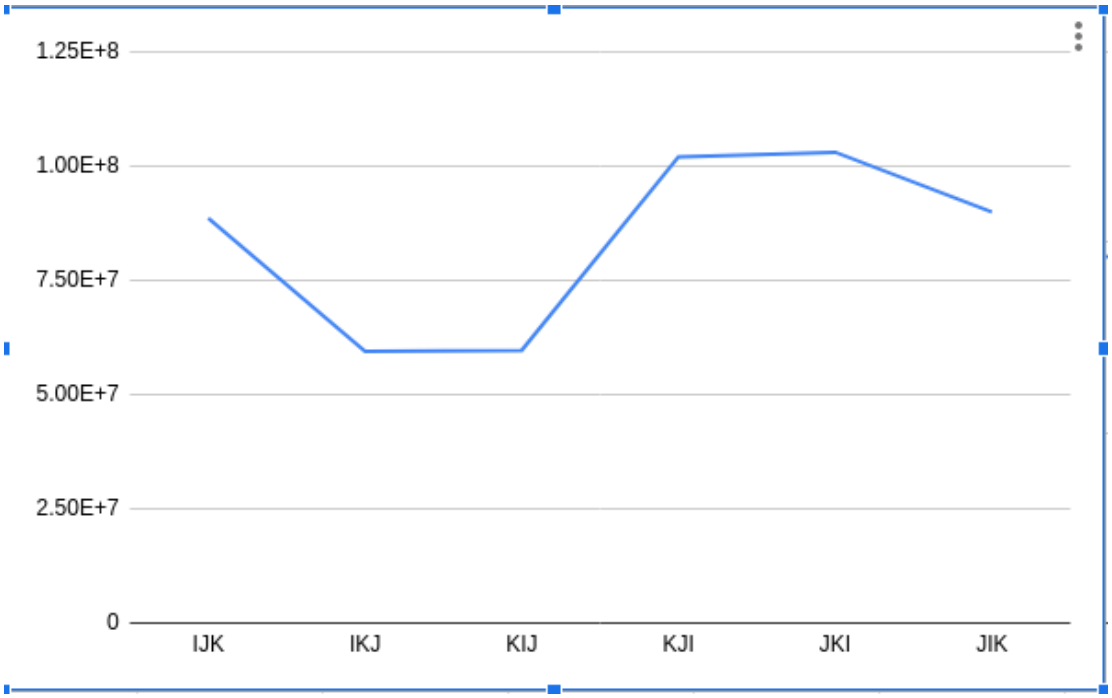
	128	**256**	**512**
IJK	10444267	88626850	724739624
IKJ	9467350	59430480	420068011
KIJ	9305010	59704946	423859587
KJI	12716718	102065793	814084227
JKI	12827583	103077558	817735952
JIK	10200303	89970305	743842431

Graphs(*showing variation of N with permutation of i, j, k***)**

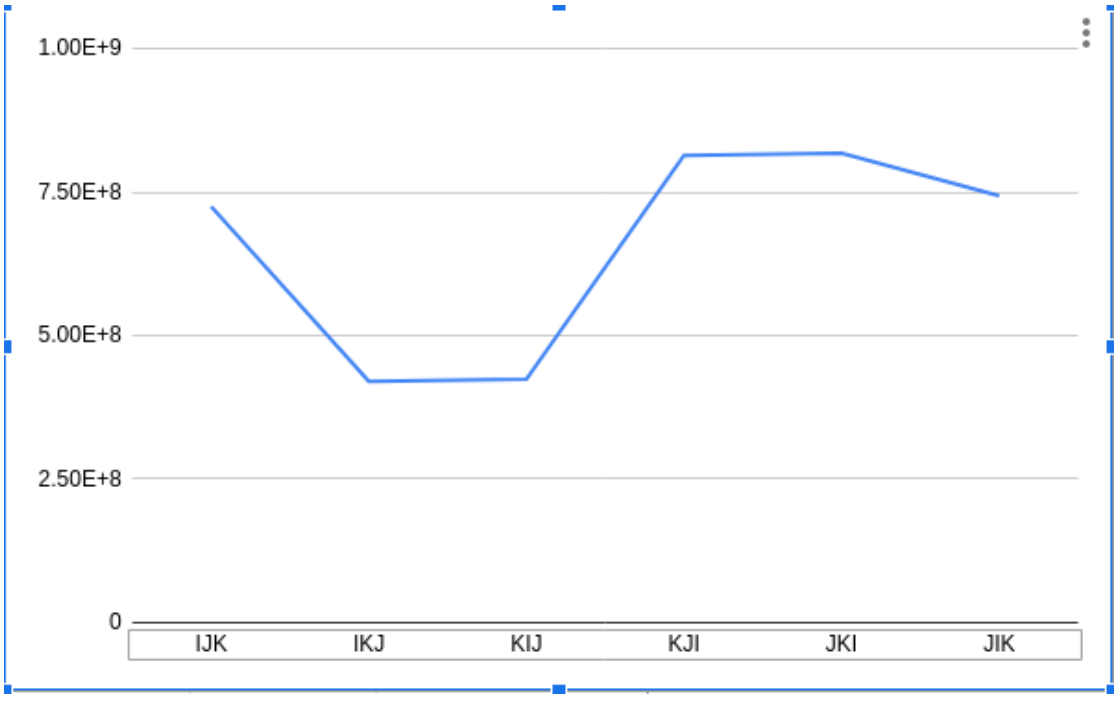
N = 128



N = 256


1.25E+8
1.00E+8
7.50E+7
5.00E+7
2.50E+7
0
IJK
IKJ
KIJ
KJI
JKI
JIK

N = 512


1.00E+9
7.50E+8
5.00E+8
2.50E+8
0
IJK
IKJ
KIJ
KJI
JKI
JIK

SUMMARY

All the observations noted here have above graphs and experimental experience as basis

1. The overall **behavior** of graph for all values of N remain **similar**
2. **Minimum** time(best sequence) = **k, i, j**
3. **Maximum** time(worst sequence) = **j, k, i**
4. The performance of **k, i, j** and **i, k , j** are very close and in most cases the difference isn't significant. Same is true with **j, k, i** and **k, j, i.**
5. Most of the performance is determined by the inner most loop:

$$Performance(as\ inner\ most\ loop):\ j\ >\ k\ >\ i$$

The above graphs are self explanatory and above were the major conclusions.