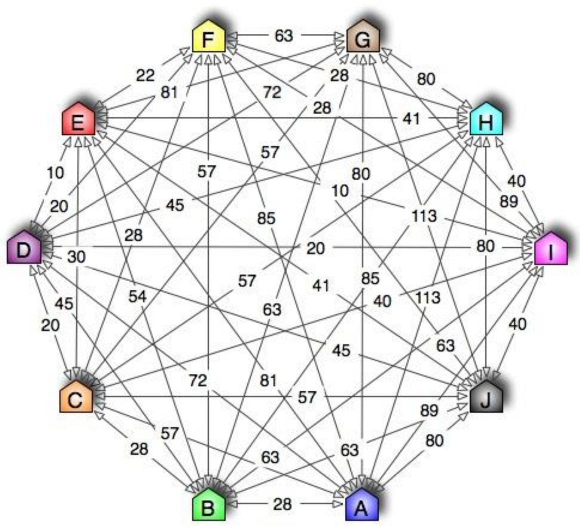
ANALYSIS of ALGORITHMS

CSE2046

ALGORITHM PROJECT REPORT



UFUK GÜRBÜZ – 150113058

AHMET GÖRÜNÜCÜ – 150113001

PROCEDURES

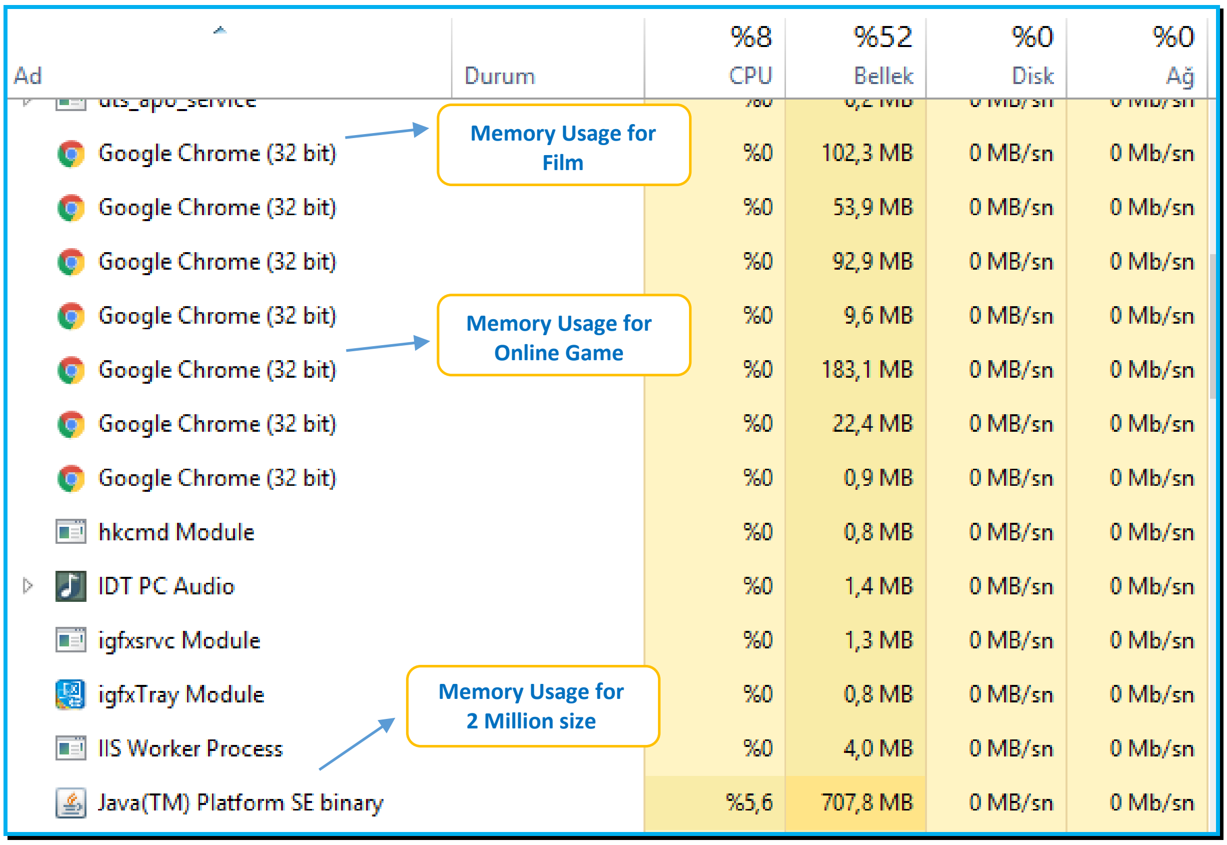
STEP 1

Designing the TSP Algorthm:

We designed a TSP algorithm which occurs three part in Java. The first part is "the nearest technic". The salesman starts from a point and goes to nearest point orderly, then the salesman returns starting point. The second part is "the changing starting point technic". The salesman starts from different points and visits other points using "the nearest technic". At the end of these two parts, the best tour is determined and this tour is saved as best tour. The final part is "optimization technic".

The algorithm takes best tour and it fixs best tour using "optimization technic". The optimiation technic makes "**probability prediction**" and changes connection of two points. Then, the algorithm calculates total distance of tour and compares with total distance of previous best tour. If the new total distance is less than old total distance, the best tour is new tour. Otherwise, the algorithm makes "**probability prediction**" again and changes other place of points.

We designed this algorithm to use less memory. Namely, you can use this algortihm for every input which has small or big size. We thinked that the memory usage is important for many user. Namely, the algorithm works in the background quietly and you can watch a film or insomuch as you can play game.



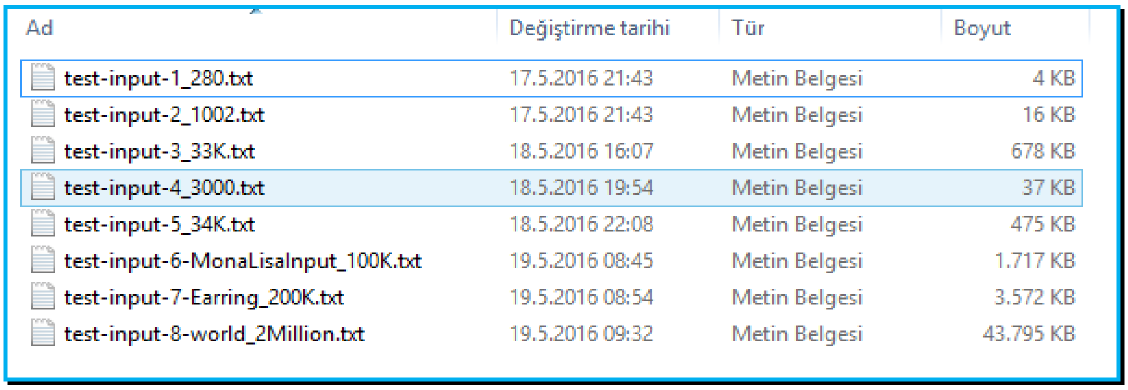
Ad	Durum	%8 CPU	%52 Bellek	%0 Disk	%0 Ağ
dts_apo_service		%0	0,2 MB	0 MB/sn	0 Mb/sn
Google Chrome (32 bit)		%0	102,3 MB	0 MB/sn	0 Mb/sn
Google Chrome (32 bit)		%0	53,9 MB	0 MB/sn	0 Mb/sn
Google Chrome (32 bit)		%0	92,9 MB	0 MB/sn	0 Mb/sn
Google Chrome (32 bit)		%0	9,6 MB	0 MB/sn	0 Mb/sn
Google Chrome (32 bit)		%0	183,1 MB	0 MB/sn	0 Mb/sn
Google Chrome (32 bit)		%0	22,4 MB	0 MB/sn	0 Mb/sn
Google Chrome (32 bit)		%0	0,9 MB	0 MB/sn	0 Mb/sn
hkcmd Module		%0	0,8 MB	0 MB/sn	0 Mb/sn
IDT PC Audio		%0	1,4 MB	0 MB/sn	0 Mb/sn
igfxsrvc Module		%0	1,3 MB	0 MB/sn	0 Mb/sn
igfxTray Module		%0	0,8 MB	0 MB/sn	0 Mb/sn
IIS Worker Process		%0	4,0 MB	0 MB/sn	0 Mb/sn
Java(TM) Platform SE binary		%5,6	707,8 MB	0 MB/sn	0 Mb/sn

STEP 2

Coding and Running:

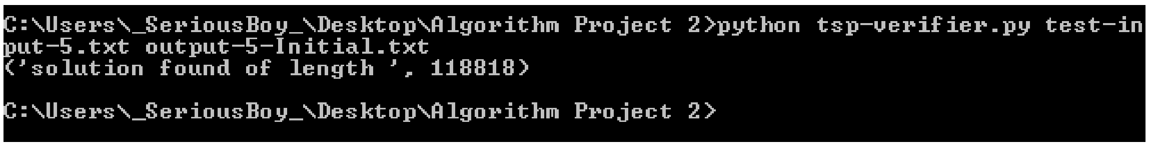
We implemented a TSP algorithm which occurs three part in Java. We were careful for complexity and clarity of algorithm when we maked coding. The algorithm should has been simple, clear but effective. Also, the algorithm could for input which has big size. So, we were careful in using variables, class architecture, reference addressing. We ran our algorithm for some little size. The result is perfect. The memory usage is very small and the approximation is very good for optimal solution.

Then we ran our algorithm for many big size. The result is almost perfect. The result can be better, but the time is very short. So, we can't take best optimal solution.



Ad	Değiştirme tarihi	Tür	Boyut
test-input-1_280.txt	17.5.2016 21:43	Metin Belgesi	4 KB
test-input-2_1002.txt	17.5.2016 21:43	Metin Belgesi	16 KB
test-input-3_33K.txt	18.5.2016 16:07	Metin Belgesi	678 KB
test-input-4_3000.txt	18.5.2016 19:54	Metin Belgesi	37 KB
test-input-5_34K.txt	18.5.2016 22:08	Metin Belgesi	475 KB
test-input-6-MonaLisaInput_100K.txt	19.5.2016 08:45	Metin Belgesi	1.717 KB
test-input-7-Earring_200K.txt	19.5.2016 08:54	Metin Belgesi	3.572 KB
test-input-8-world_2Million.txt	19.5.2016 09:32	Metin Belgesi	43.795 KB

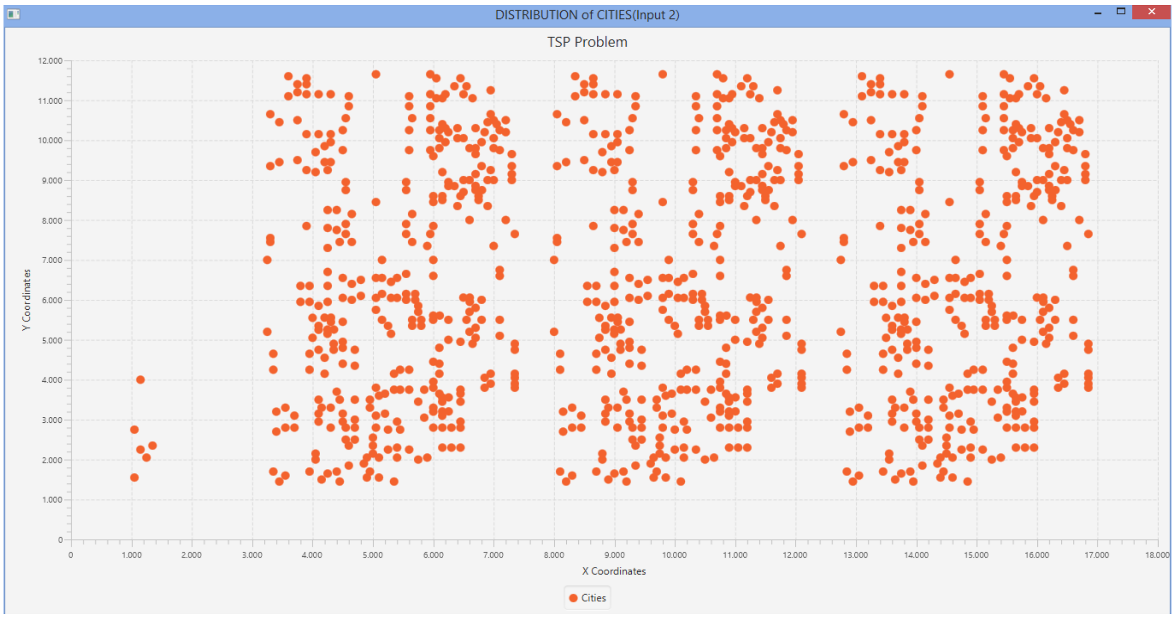
Let us examine usage of python verifier code and see example result.



```
C:\Users\_SeriousBoy_\Desktop\Algorithm Project 2>python tsp-verifier.py test-in
put-5.txt output-5-Initial.txt
('solution found of length ', 118818)

C:\Users\_SeriousBoy_\Desktop\Algorithm Project 2>
```

As we saw above, our algorithm gives correct-valid results. Also, we added a "graphic class" onto our source code. You can see place of points onto coordinate system.



STEP 3

Illustrating and Analyzing Results:

We ran and tested our algorithm for all inputs. The algorithm calculated total distance, fixed them and created best tour for all inputs. We verified all results in python verifier code and compared optimal results.

	OPTIMAL	EMPIRICAL	
INPUTS - SIZES	Total Distance	Total Distance	Memory Usage
Input 1 - 280 Cities	2579	2750	18 Mb
Input 2 - 1002 Cities	259,045	309,025	20 Mb
Input 4 - 2924 Cities	10,128	12,048	30 Mb
Input 5 - 34K Cities	97,254	118,818	45 Mb
Extra Input 6 - 100K Cities	5,757,191	6,857,709	79 Mb
Extra Input 7 - 200K Cities	8,171,677	9,742,227	150 Mb
Extra Input 8 - 2 Million Cities (Different Set)	~	17,657	709 Mb

CONCLUSION

We worked hard for solving TSP Problem and approach optimal solution. Thus, we designed a algorithm which occurs three part. The parts are "the nearest technic", "the changing starting point technic", "optimization technic".

We tested algorithm for all inputs and saved results. Then, we compared results of our friends. Our results are fine. The algorithms of some friends falt down for inputs which have big size. But, our algorithm worked smooth fort hem. Insomuch that, we tried for input which has "2 million size". Then we verified result in python verifier code. The verifier program gave correct response.

REFERENCES

- Introduction to Design and Analysis of Algorithms, Anany Levitin, 3rd Edition
- https://en.wikipedia.org/wiki/Travelling_salesman_problem
- https://www.youtube.com/watch?v=BmsC6AEbkrw
- https://www.seas.gwu.edu/~simhaweb/champalg/tsp/tsp.html
- https://class.coursera.org/algo2-002/lecture/181
- http://cs.stackexchange.com/questions/1749/dijsktras-algorithm-applied-to-travelling-salesman-problem
- Experience of Assistant Prof. Ömer Korçak ☺