CSCE 629: Analysis of Algorithms Fall 2021
Course Project Report

Kaushal Prudhvi Raj T
UIN: 630009512
November 30, 2021

Brief Description:

The project here was aimed at Network Optimization- Finding the maximum Bandwidth between two end points using undirected weighted graphs of very large number of vertices, mainly analyzing the best possible algorithms Dijkstra(O(mlogn)), Kruskal's(O(mlogn)) and providing a good reason why one has to be chosen over other in different scenarios. Also these algorithm implementations are also compared to naïve implementation of maximum Bandwidth paths which takes 0(n^2). My implementation of data structures for graphs, heaps, Kruskal's is accepted by Dr. Jianer Chen upon discussion of my implementation during office hours in person. Dr. Jianer Chen considered my implementation is low level and asked me to cite his acceptance in the report.

All the implementation here is done in C++.

1) Random Graph Generation:

An adjacency list implementation of Graph is performed by creating a very low-level implementation of structure of nodes that are linked. All the code here was done making most use of the pointers and all the dynamic allocation of memory was cleanly handled. The graph was constructed in such a way that all the vertices are connected and there exists a path between any pair of vertices.

a) Graph with a degree 6:

A random Poisson distribution of mean set to 6 is chosen and a generator randomly generates the vertices between 0 and 4999. Alongside an edge is formed, stored as node of source and destination, by joining the vertices and a random weight of highest possible randomness is assigned by seeding every time a new vertex is randomly generated. The edge count is also maintained, and the edge structure defined here has been fully utilized while sorting the edges for Kruskal's and while constructing the Maximum Spanning Tree.

b) Graph with degree of about 20%.

A dense graph is similarly generated as above now by setting the mean of Poisson distribution to 1000 and as a result every vertex is now connected to about 20% of the total number of vertices. The weights, edges are similarly created and stored as above.

2) Heap Structure:

For implementation of Dijkstra algorithm using Heaps, all the necessary functions like MAXHEAPIFY, MAXIMUM, INSERT at a particular position, HEAPSORT, DELETEMAX, DELETE any element, swap were implemented and no in-built libraries were used. Also an array called D[] is used to store the Bandwidths of vertices and another array called P[] is used to store the positions of vertices in heap. These were simultaneously modified when any changes were made to our heap. These functions were also re-used while sorting edges for Kruskal's.

3) Routing Algorithms:

a) **Dijkstra without Heap: NoHeapDijkstra(Graph, s, t)**
   a. Normal implementation of the algorithm by picking the maximum BW fringe from the available fringes and from there expanding the scope of fringes seen.
   b. The above procedure is performed for all the adjacent nodes to the source and from there to their corresponding neighbors thereby propagating till the destination(sink) is visited. The algorithm is written completely based on the adjacency list pointers and finally the result is a max Bandwidth path from Source to destination.
   c. The theoretical computational complexity yields O(V^2).

b) **Dijkstra with Heap**: **HeapDijkstra(Graph, s , t)**
   a. Here, a better way of finding a fringing with maximum BW is written using MAXIMUM function using heaps.
   b. Also the other functionalities like heapify, insert into heaps, delete max, delete at any given position, heapsort were implemented.
   c. The theoretical computational complexity yields O(ElogV).

c) **Kruskal's Algorithm**: **KruskalMST(Graph, s, t)**
   a. The Kruskal's algorithm for Maximum Spanning Tree was implemented. The methods Union by rank, Find have been implemented using arrays. Also the s-t path is found by simple Breadth First Search after constructing the Maximum Spanning Tree.

b. Here the heapsort implemented using heaps is used to sort edges based on weights in non-increasing order and hence picking the edges of maximum weight making sure that no cycle is formed while picking the edges by effectively making use of Union-Find functions.
c. The theoretical time complexity is dominated by the heapsort of edges and a significant amount for finding the s-t path after constructing the Maximum Spanning Tree but the upper bound comes out to be O(ElogV).

4) **Results and Analysis**:
   a. The source and sink vertices were chosen in a highly random manner by seeding the random function by a high degree.
   b. The tests are performed for by choosing 5 pairs of sparse and dense graphs with 5 pairs of random vertices each and the corresponding paths were also presented as output.

5) **Observations**:

a) It has been observed that in all the cases, Dijkstra without heap performs the worst as there is no method to easily determine the maximum fringe while traversing. Also, it possibly has to loop each time across the entire array of fringes to pick a maximum one which increases the time complexity
b) The Kruskal's Maximum Spanning Tree for Maximum Bandwidth Paths seemed to produce peculiar results. It competes with Dijkstra with heaps in cases of Sparse graphs, in a way that sometimes its running time is very close to that of Dijkstra with heap. In case if the random graph has very fewer edges, then Kruskal is the best performer. Hence, I can say that Kruskal is as better as Dijkstra with heap when Sparse graphs are considered. My analysis helped me understand that Kruskal can be used in networks that are lightly loaded as in a small corporate office networks and home networks
c) And the best performer in all kinds of graphs is the Dijkstra with heaps. The key element of this the easy way of storing maximum BW fringes in a heap and easily retrieving in the order of O(1) and heapifying in O(logn) makes it stand out and run very fast on all kinds of graphs proving to be the best choice among the three if the graph is very dense as in the case of real world networks which have an IP address range of O(2^32) which gives rise to the edges of the order

O((2^32C1)* (2^32)^2) which is a very dense graph and hence using Kruskal would complicate the routing by taking all edges. Hence Dijkstra with heap is a best choice in terms of real-world routing.

For Sparse graphs:
Running Time:
Best: Dijkstra with Heap
Worst: Dijkstra without heaps

Dijkstra With Heap < Kruskal's < Dijkstra Without Heaps

For Dense graphs:
Best: Dijkstra with Heap
Worst: Kruskal's

Dijkstra With Heap < Dijkstra Without Heaps < Kruskal's

Console Output:

1) For 5pairs of Graphs with 5 pairs of vertices
2) After this 5 pairs of graphs with 5 pairs of vertices for each is also presented.
3) Also a representative graph is stored on a file graph.txt that describes my adjacency list implementation.

==========

Now Testing Graphs Pair[1] with vertices (s: 4872,t:4293)

==========Graph with Degree 6 ==========
Using DIJKSTRA(no HEAP): The BW[4293]is 2971 and the path is(4293<-4395<-6<-396<-1339<-2267<-2925<-3785<-1168<-3414<-22<-4170<-1355<-4534<-2043<-3736<-1431<-4961<-2450<-2090<-1043<-3664<-3524<-3028<-3029<-1599<-4895<-4631<-2671<-331<-3474<-3819<-1366<-4250<-1277<-4872)
Practical Running TIme of Dijkstra(No Heap is):124ms

Using HEAP DIJKSTRA :The BW[4293]is 2971 and the path is(4293<-4395<-6<-396<-1339<-2267<-2925<-3785<-1168<-3414<-22<-851<-4723<-1827<-2052<-2341<-4109<-2390<-3527<-558<-2413<-2064<-4142<-1996<-1054<-4990<-1458<-2363<-2086<-410<-2685<-1629<-1628<-1820<-61<-844<-3966<-2017<-2090<-1043<-3664<-3524<-3028<-3029<-1599<-4895<-4631<-2671<-331<-3474<-3819<-1366<-4250<-1277<-4872)
Practical Running TIme of Dijkstra(With Heap is):16ms

Using KRUSKAL's: The BW[4293]is 2971 and the path is(4293<-4395<-6<-396<-1339<-2267<-2925<-3785<-1168<-3414<-22<-4170<-4864<-164<-171<-4568<-1529<-1052<-165<-3395<-4068<-1724<-2485<-

1148<-2363<-1458<-4990<-361<-4791<-2547<-3686<-2534<-350<-1461<-4938<-292<-3666<-3555<-1108<-130<-487<-2107<-2285<-2403<-1878<-3524<-3028<-3029<-1599<-4895<-4631<-2671<-331<-3474<-3819<-1366<-4250<-1277<-4872)
Practical Running TIme of Kruskal's: 31ms

==========Graph with Degree of about 20%==========
Using DIJKSTRA(no HEAP): The BW[4293]is 3182 and the path is(4293<-2805<-2443<-522<-3148<-3591<-1354<-1822<-3650<-3647<-382<-3024<-2254<-4946<-4176<-4872)
Practical Running TIme of Dijkstra(No Heap is):172ms

Using HEAP DIJKSTRA :The BW[4293]is 3182 and the path is(4293<-2805<-1499<-4698<-747<-1183<-505<-4176<-4872)
Practical Running TIme of Dijkstra(With Heap is):47ms

Using KRUSKAL's: The BW[4293]is 3182 and the path is(4293<-2805<-2443<-2884<-500<-914<-2019<-4862<-524<-227<-4199<-351<-2362<-2285<-2741<-1407<-2851<-3537<-2937<-380<-29<-1351<-1510<-2398<-209<-672<-160<-1755<-3508<-2093<-2267<-744<-3350<-3351<-4985<-324<-874<-3889<-409<-2257<-2256<-2031<-814<-1029<-4411<-1847<-1706<-3509<-1861<-2122<-2784<-2173<-1894<-1483<-4946<-4176<-4872)
Practical Running TIme of Kruskal's: 328ms
==========

==========

Now Testing Graphs Pair[2] with vertices (s: 2005,t:3402)

==========Graph with Degree 6 ==========
Using DIJKSTRA(no HEAP): The BW[3402]is 2463 and the path is(3402<-3403<-897<-896<-984<-1531<-1106<-2454<-1237<-1126<-2005)
Practical Running TIme of Dijkstra(No Heap is):125ms

Using HEAP DIJKSTRA :The BW[3402]is 2463 and the path is(3402<-3403<-1721<-3087<-1415<-1144<-4785<-1402<-1712<-572<-2454<-1237<-1126<-2005)
Practical Running TIme of Dijkstra(With Heap is):15ms

Using KRUSKAL's: The BW[3402]is 2463 and the path is(3402<-4973<-4544<-319<-318<-2355<-3684<-534<-1142<-4275<-2615<-3391<-2284<-195<-2451<-3710<-2219<-509<-4623<-4437<-165<-2727<-4507<-4506<-4187<-1662<-601<-2695<-1199<-3267<-3464<-2335<-1800<-4035<-3282<-4041<-4040<-4972<-688<-144<-3075<-375<-1378<-1713<-1127<-1126<-2005)
Practical Running TIme of Kruskal's: 32ms

==========Graph with Degree of about 20%==========
Using DIJKSTRA(no HEAP): The BW[3402]is 3283 and the path is(3402<-535<-2931<-942<-2953<-4620<-560<-1210<-2182<-2213<-951<-4818<-895<-894<-1196<-3133<-4733<-2818<-4003<-4349<-4598<-2295<-4939<-2005)
Practical Running TIme of Dijkstra(No Heap is):172ms

Using HEAP DIJKSTRA :The BW[3402]is 3283 and the path is(3402<-755<-487<-1403<-1017<-2236<-238<-1484<-545<-672<-3840<-959<-4349<-4598<-2295<-4939<-2005)
Practical Running TIme of Dijkstra(With Heap is):63ms

Using KRUSKAL's: The BW[3402]is 3283 and the path is(3402<-535<-2931<-1376<-2067<-3278<-3232<-96<-1752<-2442<-33<-3699<-4550<-4823<-180<-4307<-4502<-4311<-4193<-3491<-1460<-2042<-4472<-533<-1656<-2549<-1359<-250<-662<-663<-4145<-3418<-635<-4547<-2524<-3287<-4912<-1718<-193<-3410<-1529<-2611<-3184<-1702<-602<-1034<-3400<-319<-4124<-2182<-1210<-560<-4620<-2953<-2427<-4993<-2145<-1455<-1065<-2091<-2058<-2489<-1430<-4598<-2295<-4939<-2005)
Practical Running TIme of Kruskal's: 359ms
==========

==========

Now Testing Graphs Pair[3] with vertices (s: 2109,t:1694)

==========Graph with Degree 6 ==========
Using DIJKSTRA(no HEAP): The BW[1694]is 2779 and the path is(1694<-352<-393<-3720<-1071<-4964<-3101<-3708<-1901<-2352<-4492<-563<-1387<-1386<-4999<-2498<-665<-373<-2086<-3686<-1303<-1629<-3754<-1337<-2772<-262<-4604<-4752<-2958<-3622<-2797<-257<-1325<-73<-4890<-893<-592<-4317<-4149<-2877<-384<-2895<-3986<-3987<-2109)
Practical Running TIme of Dijkstra(No Heap is):125ms

Using HEAP DIJKSTRA :The BW[1694]is 2779 and the path is(1694<-352<-393<-3720<-1071<-4964<-3101<-3708<-1901<-2352<-3790<-2478<-2341<-3462<-3025<-172<-3397<-22<-1470<-1621<-2374<-2313<-3534<-1627<-2452<-3683<-442<-443<-3656<-1303<-1629<-3754<-1337<-2772<-262<-4604<-4752<-2958<-3622<-2797<-257<-1325<-73<-4890<-893<-592<-4317<-4149<-2877<-384<-2895<-3986<-3987<-2109)
Practical Running TIme of Dijkstra(With Heap is):31ms

Using KRUSKAL's: The BW[1694]is 2779 and the path is(1694<-352<-393<-3720<-1071<-4964<-3101<-3307<-1982<-2441<-2618<-3180<-4137<-2190<-2389<-3996<-4312<-4313<-1566<-3683<-442<-443<-3656<-1303<-1629<-3754<-1337<-2772<-262<-4604<-4752<-2958<-3622<-2797<-257<-1325<-73<-4890<-893<-592<-4317<-4149<-2877<-384<-2895<-3986<-3987<-2109)
Practical Running TIme of Kruskal's: 32ms

==========Graph with Degree of about 20%==========
Using DIJKSTRA(no HEAP): The BW[1694]is 3274 and the path is(1694<-1715<-1072<-2879<-4944<-3476<-1013<-335<-2586<-4961<-217<-2500<-161<-3300<-1017<-4242<-2860<-2353<-2352<-4430<-4557<-1684<-2807<-261<-180<-2926<-1993<-2109)
Practical Running TIme of Dijkstra(No Heap is):172ms

Using HEAP DIJKSTRA :The BW[1694]is 3274 and the path is(1694<-1715<-4261<-1490<-2184<-480<-1673<-3261<-549<-766<-1614<-1383<-2425<-579<-2807<-261<-180<-2926<-1993<-2109)
Practical Running TIme of Dijkstra(With Heap is):46ms

Using KRUSKAL's: The BW[1694]is 3274 and the path is(1694<-1715<-758<-151<-1749<-2387<-1945<-1112<-1023<-2044<-4086<-4087<-3718<-1431<-1572<-2511<-2505<-3882<-1939<-2751<-4556<-1562<-2880<-1805<-1383<-2425<-579<-2807<-261<-180<-2926<-1993<-2109)
Practical Running TIme of Kruskal's: 344ms
==========

==========

Now Testing Graphs Pair[4] with vertices (s: 4455,t:2021)

======================Graph with Degree 6 ======================
Using DIJKSTRA(no HEAP): The BW[2021]is 2597 and the path is(2021<-3853<-3237<-2585<-4570<-2345<-2194<-927<-3337<-3336<-3406<-3286<-1228<-1112<-4455)
Practical Running TIme of Dijkstra(No Heap is):125ms

Using HEAP DIJKSTRA :The BW[2021]is 2597 and the path is(2021<-1426<-1232<-1763<-2130<-951<-4890<-453<-2922<-3807<-3806<-4648<-4888<-751<-3383<-3052<-2143<-1674<-1675<-2279<-3286<-1228<-1112<-4455)
Practical Running TIme of Dijkstra(With Heap is):16ms

Using KRUSKAL's: The BW[2021]is 2597 and the path is(2021<-3853<-723<-1402<-2472<-4636<-664<-3917<-1482<-2124<-3156<-1250<-1129<-3361<-3791<-355<-4699<-1168<-306<-4254<-677<-2014<-1881<-369<-2344<-3271<-2261<-3657<-3656<-1396<-4923<-1992<-2609<-4187<-244<-38<-3440<-4603<-452<-4237<-4158<-4599<-3000<-112<-631<-45<-2964<-2279<-3286<-1228<-1112<-4455)
Practical Running TIme of Kruskal's: 47ms

==================Graph with Degree of about 20%====================
Using DIJKSTRA(no HEAP): The BW[2021]is 3205 and the path is(2021<-2425<-560<-3031<-1822<-1939<-4050<-3005<-200<-480<-295<-1714<-4672<-1306<-1937<-2083<-4310<-2174<-2908<-4282<-406<-3569<-3451<-4455)
Practical Running TIme of Dijkstra(No Heap is):187ms

Using HEAP DIJKSTRA :The BW[2021]is 3205 and the path is(2021<-2425<-560<-3031<-1822<-1939<-4050<-3005<-200<-480<-295<-1714<-4672<-1306<-1937<-2083<-4310<-2174<-2908<-4282<-406<-3569<-3451<-4455)
Practical Running TIme of Dijkstra(With Heap is):63ms

Using KRUSKAL's: The BW[2021]is 3205 and the path is(2021<-2425<-560<-3031<-1822<-1939<-4050<-3005<-200<-480<-295<-1714<-4672<-1306<-1937<-2083<-4310<-2174<-2908<-4282<-406<-3569<-3451<-4455)
Practical Running TIme of Kruskal's: 353ms
==========

==========

Now Testing Graphs Pair[5] with vertices (s: 2509,t:162)

======================Graph with Degree 6 ======================
Using DIJKSTRA(no HEAP): The BW[162]is 2657 and the path is(162<-4285<-2190<-3279<-3176<-4855<-4748<-3899<-808<-3730<-184<-185<-4528<-4529<-1385<-2729<-3591<-3253<-3045<-4983<-2738<-1129<-3195<-3160<-1624<-95<-1078<-1481<-4616<-3616<-2391<-3225<-4793<-1418<-4831<-4816<-948<-3510<-1939<-1428<-3819<-3818<-3858<-4193<-984<-3893<-3022<-1912<-433<-118<-636<-4118<-2509)
Practical Running TIme of Dijkstra(No Heap is):125ms

Using HEAP DIJKSTRA :The BW[162]is 2657 and the path is(162<-4285<-2190<-3279<-3176<-4855<-4748<-3899<-808<-3730<-184<-185<-4528<-4529<-1385<-2729<-3591<-3253<-2782<-2743<-3460<-1631<-613<-3160<-1624<-95<-1078<-1481<-4616<-3616<-2391<-3225<-4793<-1418<-4831<-4816<-948<-3510<-1939<-1428<-3819<-3818<-3858<-4193<-984<-3893<-3022<-1912<-433<-118<-636<-4118<-2509)
Practical Running TIme of Dijkstra(With Heap is):31ms

Using KRUSKAL's: The BW[162]is 2657 and the path is(162<-4285<-2190<-3279<-3176<-4855<-4748<-3899<-808<-3730<-184<-185<-4528<-4529<-1385<-2729<-3591<-3253<-3045<-4983<-2738<-1129<-3195<-3160<-1624<-95<-1078<-1481<-4616<-3616<-2391<-3225<-4793<-1418<-4831<-4816<-948<-3510<-1939<-1428<-3819<-3818<-3858<-4193<-984<-3893<-3022<-1912<-433<-118<-636<-4118<-2509)
Practical Running TIme of Kruskal's: 47ms

==========Graph with Degree of about 20%==========
Using DIJKSTRA(no HEAP): The BW[162]is 3297 and the path is(162<-4333<-567<-2892<-4310<-2175<-4484<-427<-501<-2402<-4693<-4386<-604<-4350<-1811<-3714<-2745<-2064<-2248<-2509)
Practical Running TIme of Dijkstra(No Heap is):172ms

Using HEAP DIJKSTRA :The BW[162]is 3297 and the path is(162<-4333<-567<-2892<-4310<-2175<-4484<-427<-501<-2402<-4693<-4386<-604<-4350<-1811<-3714<-2745<-2064<-2248<-2509)
Practical Running TIme of Dijkstra(With Heap is):63ms

Using KRUSKAL's: The BW[162]is 3297 and the path is(162<-4333<-567<-2892<-4310<-2175<-4484<-427<-501<-2402<-4693<-4386<-604<-4350<-1811<-3714<-2745<-2064<-2248<-2509)
Practical Running TIme of Kruskal's: 343ms
==========

Console Output: For 5 pairs of graphs with 5 pairs of vertices for each pair of sparse and dense graph

Now Testing Pair[1] Graph

==========

Now Testing using Pair[1] with vertices (s: 3734,t:4907)

==========Graph with Degree 6 ==========
Using DIJKSTRA(no HEAP): The BW[4907]is 2949 and the path is(4907<-655<-3651<-4534<-3010<-3352<-4757<-4909<-837<-1978<-144<-4092<-3066<-3728<-2050<-888<-1979<-4752<-2617<-338<-1879<-481<-371<-4555<-1200<-2047<-3734)
Practical Running TIme of Dijkstra(No Heap is):120ms

Using HEAP DIJKSTRA :The BW[4907]is 2949 and the path is(4907<-655<-3651<-3207<-2040<-771<-4994<-4351<-70<-1394<-4731<-2796<-4172<-2254<-1020<-4074<-2498<-4197<-338<-1879<-481<-371<-4555<-1200<-2047<-3734)
Practical Running TIme of Dijkstra(With Heap is):13ms

Using KRUSKAL's: The BW[4907]is 2949 and the path is(4907<-655<-353<-352<-4710<-4711<-3383<-4980<-4368<-787<-1367<-1366<-3362<-2653<-3665<-3017<-3372<-2395<-440<-2974<-2032<-3705<-4498<-1151<-2455<-1356<-3866<-3137<-2196<-2197<-4187<-915<-3750<-2460<-1620<-4789<-1816<-3071<-1162<-2213<-4964<-2621<-550<-2692<-3352<-4757<-4909<-837<-1978<-144<-4092<-3066<-3728<-4416<-805<-3591<-2886<-165<-4674<-2349<-1800<-4867<-4752<-2617<-338<-1879<-481<-371<-4555<-1200<-2047<-3734)
Practical Running TIme of Kruskal's: 77ms

==========Graph with Degree of about 20%==========
Using DIJKSTRA(no HEAP): The BW[4907]is 3281 and the path is(4907<-829<-1622<-4416<-2562<-3491<-844<-1205<-1756<-623<-949<-792<-2549<-4524<-2865<-3230<-958<-912<-487<-3368<-3352<-1810<-522<-4446<-870<-3830<-3734)
Practical Running TIme of Dijkstra(No Heap is):221ms

Using HEAP DIJKSTRA :The BW[4907]is 3281 and the path is(4907<-829<-1622<-4416<-2562<-3491<-844<-1205<-1756<-623<-949<-792<-2549<-4524<-2865<-3230<-958<-912<-487<-3368<-3352<-1810<-522<-4446<-870<-3830<-3734)
Practical Running TIme of Dijkstra(With Heap is):55ms

Using KRUSKAL's: The BW[4907]is 3281 and the path is(4907<-829<-1622<-4416<-2562<-3491<-844<-1205<-1756<-623<-949<-792<-2549<-4524<-2865<-3230<-958<-912<-487<-3368<-3352<-1810<-522<-4446<-870<-3830<-3734)
Practical Running TIme of Kruskal's: 358ms
==========

==========

Now Testing using Pair[2] with vertices (s: 299,t:2512)

==========Graph with Degree 6 ==========
Using DIJKSTRA(no HEAP): The BW[2512]is 3045 and the path is(2512<-3762<-3763<-3215<-898<-97<-1677<-1412<-91<-2381<-1194<-4661<-2727<-1100<-1789<-4351<-4727<-4237<-4430<-1853<-3891<-4530<-625<-2519<-4302<-1748<-2303<-146<-2073<-4579<-2353<-853<-1229<-299)
Practical Running TIme of Dijkstra(No Heap is):127ms

Using HEAP DIJKSTRA :The BW[2512]is 3045 and the path is(2512<-3762<-3763<-3215<-898<-97<-1677<-1412<-91<-2381<-1194<-4661<-2727<-1100<-1789<-4351<-4727<-4237<-4430<-1853<-3891<-4530<-625<-2519<-4302<-1748<-2303<-146<-2073<-4579<-2353<-853<-1229<-299)
Practical Running TIme of Dijkstra(With Heap is):22ms

Using KRUSKAL's: The BW[2512]is 3045 and the path is(2512<-3762<-3763<-3215<-898<-97<-1677<-22<-286<-3910<-1911<-4378<-3841<-2703<-365<-1475<-4144<-1297<-1296<-4783<-2425<-2864<-4594<-4283<-378<-2133<-752<-873<-2824<-4867<-4866<-2514<-2515<-1256<-3291<-4536<-4530<-625<-2519<-4302<-1748<-2303<-146<-2073<-4579<-2353<-853<-1229<-299)
Practical Running TIme of Kruskal's: 64ms

==========Graph with Degree of about 20%==========
Using DIJKSTRA(no HEAP): The BW[2512]is 3240 and the path is(2512<-1294<-452<-1143<-1707<-274<-299)
Practical Running TIme of Dijkstra(No Heap is):172ms

Using HEAP DIJKSTRA :The BW[2512]is 3240 and the path is(2512<-1294<-4136<-2875<-36<-2902<-1771<-1561<-253<-2499<-1852<-2608<-891<-304<-493<-4993<-749<-3159<-1955<-1831<-4665<-1707<-274<-299)
Practical Running TIme of Dijkstra(With Heap is):58ms

Using KRUSKAL's: The BW[2512]is 3240 and the path is(2512<-1294<-4136<-2875<-2240<-1154<-4060<-1335<-433<-1977<-3787<-855<-1018<-382<-610<-442<-1005<-4663<-3416<-1506<-4885<-634<-585<-1002<-4602<-155<-15<-1406<-389<-106<-1096<-1308<-3353<-2282<-1483<-478<-183<-2574<-692<-2081<-

3706<-989<-2480<-4694<-3727<-541<-4760<-1522<-1764<-893<-4313<-1097<-3099<-13<-4285<-3553<-1684<-1707<-274<-299)
Practical Running TIme of Kruskal's: 353ms
==========

==========

Now Testing using Pair[3] with vertices (s: 876,t:2451)

==========Graph with Degree 6 ==========
Using DIJKSTRA(no HEAP): The BW[2451]is 2927 and the path is(2451<-2434<-2446<-4394<-1481<-857<-2656<-3676<-2498<-459<-2195<-713<-2249<-2740<-4650<-876)
Practical Running TIme of Dijkstra(No Heap is):124ms

Using HEAP DIJKSTRA :The BW[2451]is 2927 and the path is(2451<-1769<-1881<-2223<-3177<-1632<-303<-302<-2479<-2415<-3676<-2498<-459<-4206<-719<-4411<-2327<-745<-1318<-996<-4822<-616<-2249<-2740<-4650<-876)
Practical Running TIme of Dijkstra(With Heap is):15ms

Using KRUSKAL's: The BW[2451]is 2927 and the path is(2451<-2450<-2984<-4957<-2280<-527<-1447<-4991<-2706<-530<-159<-4002<-1575<-4762<-4221<-62<-1726<-4800<-325<-2740<-4650<-876)
Practical Running TIme of Kruskal's: 33ms

==========Graph with Degree of about 20%==========
Using DIJKSTRA(no HEAP): The BW[2451]is 3261 and the path is(2451<-1173<-2729<-3583<-1729<-4610<-1018<-2012<-1705<-1914<-259<-2875<-320<-3529<-3919<-3858<-2916<-387<-4246<-4560<-3411<-2223<-3068<-876)
Practical Running TIme of Dijkstra(No Heap is):203ms

Using HEAP DIJKSTRA :The BW[2451]is 3261 and the path is(2451<-1173<-2729<-3583<-1729<-4610<-1018<-1587<-3103<-4537<-1442<-2245<-3579<-4609<-1460<-854<-4964<-2092<-917<-353<-1282<-4085<-1172<-2518<-3716<-1680<-2916<-387<-4246<-4560<-3411<-2223<-3068<-876)
Practical Running TIme of Dijkstra(With Heap is):75ms

Using KRUSKAL's: The BW[2451]is 3261 and the path is(2451<-1173<-2729<-3583<-1729<-4610<-1018<-1587<-291<-4869<-1005<-2766<-869<-2832<-3049<-440<-371<-2662<-924<-2649<-723<-410<-2298<-732<-1631<-2001<-3317<-3459<-187<-1540<-3982<-2680<-4773<-4664<-903<-2527<-3085<-534<-4892<-2519<-2533<-2064<-1292<-4929<-812<-259<-2875<-320<-3529<-3919<-3858<-2916<-387<-4246<-4560<-3411<-2223<-3068<-876)
Practical Running TIme of Kruskal's: 350ms
==========

==========

Now Testing using Pair[4] with vertices (s: 155,t:4992)

==========Graph with Degree 6 ==========

Using DIJKSTRA(no HEAP): The BW[4992]is 3001 and the path is(4992<-847<-3427<-444<-3022<-1500<-497<-2819<-4545<-1875<-1821<-3575<-2133<-4026<-4027<-1752<-3470<-4209<-2183<-3733<-743<-3915<-3914<-155)
Practical Running TIme of Dijkstra(No Heap is):127ms

Using HEAP DIJKSTRA :The BW[4992]is 3001 and the path is(4992<-847<-3427<-444<-3022<-1500<-1501<-4508<-478<-196<-2802<-4786<-1971<-1038<-2619<-2618<-4753<-774<-1932<-1586<-4762<-3068<-2321<-745<-1152<-354<-4110<-4452<-3585<-655<-47<-1150<-2909<-3093<-1414<-970<-1891<-1890<-4243<-3732<-3733<-743<-3915<-3914<-155)
Practical Running TIme of Dijkstra(With Heap is):20ms

Using KRUSKAL's: The BW[4992]is 3001 and the path is(4992<-847<-3427<-444<-3022<-1500<-497<-2819<-4545<-1875<-1821<-3575<-2133<-4026<-4027<-1752<-3349<-4862<-1841<-3770<-3895<-3894<-613<-501<-3176<-2341<-1483<-1923<-4537<-3372<-878<-4285<-2249<-1636<-89<-1527<-4540<-752<-3068<-2321<-745<-1152<-354<-2565<-4559<-3679<-4689<-1382<-317<-3986<-2579<-791<-2190<-1747<-3639<-3151<-3150<-1621<-4717<-76<-2095<-4313<-2585<-1230<-2827<-3253<-1384<-4012<-1271<-2909<-3093<-1414<-970<-1891<-1890<-4243<-3732<-3733<-743<-3915<-3914<-155)
Practical Running TIme of Kruskal's: 48ms

==========Graph with Degree of about 20%==========
Using DIJKSTRA(no HEAP): The BW[4992]is 3303 and the path is(4992<-2776<-1718<-1447<-1240<-1241<-535<-2422<-706<-614<-2704<-153<-709<-3297<-915<-785<-3538<-3852<-615<-1562<-4064<-2834<-512<-435<-350<-378<-426<-3332<-3995<-4251<-3886<-241<-579<-1023<-905<-164<-4112<-1213<-3446<-1847<-4612<-3875<-815<-4849<-3477<-974<-1915<-1045<-1657<-4711<-4822<-501<-3139<-2501<-4572<-3084<-3543<-3741<-1081<-1798<-3453<-4748<-2273<-47<-155)
Practical Running TIme of Dijkstra(No Heap is):184ms

Using HEAP DIJKSTRA :The BW[4992]is 3303 and the path is(4992<-2776<-1718<-1447<-1240<-750<-759<-1977<-1821<-324<-1558<-3571<-792<-3131<-2425<-3221<-3067<-1222<-4589<-1182<-1928<-4918<-2441<-964<-4483<-866<-1483<-2918<-4125<-1023<-905<-164<-4112<-1213<-3446<-1847<-4612<-3875<-815<-4849<-3477<-974<-1915<-1045<-1657<-4711<-4822<-501<-3139<-2501<-4572<-3084<-3543<-3741<-1081<-1798<-3453<-4748<-2273<-47<-155)
Practical Running TIme of Dijkstra(With Heap is):66ms

Using KRUSKAL's: The BW[4992]is 3303 and the path is(4992<-2776<-1718<-1447<-1243<-2285<-1516<-1808<-1324<-3480<-3474<-2104<-1012<-4108<-3645<-318<-756<-2725<-1473<-1577<-2746<-348<-897<-1078<-3073<-547<-646<-1566<-361<-360<-2834<-4064<-4147<-1355<-1120<-705<-3273<-2184<-125<-170<-4372<-16<-4483<-866<-1483<-2918<-4125<-1023<-905<-164<-4112<-1213<-3446<-1847<-4612<-3875<-815<-4849<-3477<-974<-1915<-1045<-1657<-4711<-4822<-501<-3139<-2501<-4572<-3084<-3543<-3741<-1081<-1798<-3453<-4748<-2273<-47<-155)
Practical Running TIme of Kruskal's: 353ms
==========

==========

Now Testing using Pair[5] with vertices (s: 1435,t:1602)

==========Graph with Degree 6 ==========
Using DIJKSTRA(no HEAP): The BW[1602]is 2971 and the path is(1602<-717<-1013<-2546<-2416<-3662<-2955<-4676<-2790<-3870<-2032<-4002<-2514<-730<-4039<-2856<-1850<-4233<-3765<-3764<-1826<-1827<-3433<-1791<-3834<-1408<-2338<-2339<-994<-4249<-2802<-882<-8<-2753<-1787<-2980<-2995<-1435)
Practical Running TIme of Dijkstra(No Heap is):130ms

Using HEAP DIJKSTRA :The BW[1602]is 2971 and the path is(1602<-717<-1013<-2546<-2416<-3662<-2955<-4676<-2704<-1604<-2889<-2261<-4407<-487<-3763<-2978<-3804<-4549<-425<-1827<-1826<-3764<-3765<-4233<-1850<-2856<-4039<-4490<-3275<-1589<-4171<-1873<-3247<-1359<-864<-3523<-353<-4479<-2209<-3664<-97<-2725<-115<-1969<-882<-8<-2753<-1787<-2980<-2995<-1435)
Practical Running TIme of Dijkstra(With Heap is):23ms

Using KRUSKAL's: The BW[1602]is 2971 and the path is(1602<-717<-1013<-2546<-2416<-3662<-2955<-4676<-2704<-4480<-4582<-2698<-3455<-3639<-41<-4102<-2641<-3419<-3238<-339<-2485<-2601<-4471<-2396<-328<-404<-405<-1397<-1518<-3527<-613<-666<-1077<-1707<-2122<-1168<-872<-4276<-3757<-2903<-3516<-781<-2611<-818<-1042<-1969<-882<-8<-2753<-1787<-2980<-2995<-1435)
Practical Running TIme of Kruskal's: 41ms

==========Graph with Degree of about 20%==========
Using DIJKSTRA(no HEAP): The BW[1602]is 3284 and the path is(1602<-282<-3689<-2885<-1151<-4232<-2735<-3880<-881<-2770<-2894<-2493<-2917<-4961<-3406<-2865<-4696<-780<-2200<-1435)
Practical Running TIme of Dijkstra(No Heap is):205ms

Using HEAP DIJKSTRA :The BW[1602]is 3284 and the path is(1602<-282<-4531<-2304<-30<-2074<-771<-18<-1201<-1016<-4180<-2565<-1150<-1845<-2288<-3246<-3342<-4822<-841<-3809<-1406<-3884<-3681<-4208<-2917<-4961<-3406<-2865<-4696<-780<-2200<-1435)
Practical Running TIme of Dijkstra(With Heap is):63ms

Using KRUSKAL's: The BW[1602]is 3284 and the path is(1602<-578<-684<-608<-2127<-4170<-1651<-4770<-3830<-57<-3632<-546<-1143<-3168<-1150<-2565<-4350<-349<-262<-2315<-529<-3515<-3446<-4896<-1897<-73<-1652<-1174<-881<-2770<-2894<-2493<-2917<-4961<-3406<-2865<-4696<-780<-2200<-1435)
Practical Running TIme of Kruskal's: 363ms
==========

Now Testing Pair[2] Graph

==========

Now Testing using Pair[1] with vertices (s: 1021,t:4936)

==========Graph with Degree 6 ==========
Using DIJKSTRA(no HEAP): The BW[4936]is 2981 and the path is(4936<-873<-1787<-1160<-530<-555<-3756<-4770<-4548<-1383<-1099<-1939<-596<-4439<-1021)
Practical Running TIme of Dijkstra(No Heap is):125ms

Using HEAP DIJKSTRA :The BW[4936]is 2981 and the path is(4936<-873<-1787<-1160<-530<-555<-3756<-4770<-4548<-1383<-1099<-1939<-596<-4439<-1021)
Practical Running TIme of Dijkstra(With Heap is):13ms

Using KRUSKAL's: The BW[4936]is 2981 and the path is(4936<-873<-1787<-1160<-530<-555<-3756<-4770<-4548<-1383<-1099<-1939<-596<-4439<-1021)
Practical Running TIme of Kruskal's: 30ms

==========Graph with Degree of about 20%==========
Using DIJKSTRA(no HEAP): The BW[4936]is 3281 and the path is(4936<-1812<-2894<-2662<-1190<-4142<-4677<-218<-3776<-3663<-2488<-217<-182<-733<-3758<-2979<-361<-2832<-535<-785<-4646<-591<-801<-979<-2700<-38<-121<-360<-547<-2509<-1443<-2690<-3480<-721<-2715<-1401<-2965<-315<-1021)

Practical Running TIme of Dijkstra(No Heap is):196ms

Using HEAP DIJKSTRA :The BW[4936]is 3281 and the path is(4936<-1812<-2894<-2662<-1190<-4142<-4677<-218<-2078<-3133<-3765<-1187<-1358<-66<-2181<-1660<-1400<-261<-2715<-1401<-2965<-315<-1021)
Practical Running TIme of Dijkstra(With Heap is):55ms

Using KRUSKAL's: The BW[4936]is 3281 and the path is(4936<-1812<-2894<-2662<-1190<-4142<-4677<-218<-2078<-3133<-3765<-1187<-1358<-66<-4995<-4588<-4973<-1387<-2130<-4436<-4641<-1107<-1303<-126<-619<-2530<-247<-2690<-3480<-721<-2715<-1401<-2965<-315<-1021)
Practical Running TIme of Kruskal's: 378ms
==========

==========

Now Testing using Pair[2] with vertices (s: 2788,t:2063)

==========Graph with Degree 6 ==========
Using DIJKSTRA(no HEAP): The BW[2063]is 2976 and the path is(2063<-3525<-2734<-2014<-662<-2730<-659<-2540<-4131<-2535<-4742<-4559<-454<-2292<-3114<-2470<-504<-4184<-844<-950<-4768<-4769<-2788)
Practical Running TIme of Dijkstra(No Heap is):125ms

Using HEAP DIJKSTRA :The BW[2063]is 2976 and the path is(2063<-3525<-2734<-2735<-4364<-4105<-3839<-773<-176<-4963<-1450<-4655<-3631<-3348<-2664<-305<-1426<-1794<-3446<-4862<-4995<-828<-2009<-4378<-2602<-2417<-2054<-1891<-415<-3114<-2470<-504<-4184<-844<-950<-4768<-4769<-2788)
Practical Running TIme of Dijkstra(With Heap is):19ms

Using KRUSKAL's: The BW[2063]is 2976 and the path is(2063<-3525<-2734<-2014<-3451<-1653<-1801<-792<-4714<-451<-3967<-428<-1291<-1596<-1323<-3607<-4418<-1322<-1341<-4598<-2561<-1654<-261<-254<-1569<-1504<-3088<-2109<-2108<-4167<-3883<-4705<-203<-4995<-828<-2009<-4378<-2602<-2417<-2054<-1891<-415<-3114<-2470<-504<-4184<-844<-950<-4768<-4769<-2788)
Practical Running TIme of Kruskal's: 52ms

==========Graph with Degree of about 20%==========
Using DIJKSTRA(no HEAP): The BW[2063]is 3285 and the path is(2063<-1995<-3785<-2979<-2904<-4345<-1229<-4386<-4579<-2423<-3156<-551<-3620<-2788)
Practical Running TIme of Dijkstra(No Heap is):176ms

Using HEAP DIJKSTRA :The BW[2063]is 3285 and the path is(2063<-1995<-3785<-2979<-2904<-4345<-1229<-2953<-1709<-2634<-2805<-3505<-4392<-2678<-1334<-1341<-320<-4186<-2577<-2205<-54<-4663<-273<-4692<-1578<-2276<-4916<-2355<-233<-2724<-3101<-2317<-329<-4014<-2477<-1293<-3249<-2332<-3017<-3407<-3695<-3205<-4975<-1702<-2630<-1410<-3404<-693<-756<-3660<-2140<-3506<-2139<-551<-3620<-2788)
Practical Running TIme of Dijkstra(With Heap is):94ms

Using KRUSKAL's: The BW[2063]is 3285 and the path is(2063<-1995<-3785<-2979<-2904<-4345<-1229<-4386<-4592<-3206<-3387<-3615<-524<-4284<-1751<-3395<-548<-518<-2881<-3190<-3454<-438<-2738<-2578<-4327<-3103<-3102<-3239<-2532<-2139<-551<-3620<-2788)
Practical Running TIme of Kruskal's: 366ms
==========

==========

Now Testing using Pair[3] with vertices (s: 1852,t:1850)

=====================Graph with Degree 6 =====================
Using DIJKSTRA(no HEAP): The BW[1850]is 2621 and the path is(1850<-3421<-1552<-4365<-512<-513<-403<-866<-2948<-2949<-2383<-577<-1601<-2720<-3225<-1903<-4415<-575<-4116<-4294<-308<-2773<-4800<-4069<-2113<-763<-3845<-314<-965<-2796<-4815<-2521<-3360<-286<-741<-740<-2738<-2093<-297<-296<-4113<-4112<-73<-1852)
Practical Running TIme of Dijkstra(No Heap is):131ms

Using HEAP DIJKSTRA :The BW[1850]is 2621 and the path is(1850<-3421<-1552<-4365<-512<-513<-403<-866<-2948<-2949<-2383<-577<-1601<-2720<-3225<-1903<-4415<-575<-4116<-4294<-308<-2773<-4800<-4069<-2113<-763<-3845<-314<-965<-2796<-4815<-2521<-3360<-286<-741<-740<-2738<-2093<-297<-296<-4113<-4112<-73<-1852)
Practical Running TIme of Dijkstra(With Heap is):20ms

Using KRUSKAL's: The BW[1850]is 2621 and the path is(1850<-3421<-1552<-4365<-512<-513<-403<-866<-2948<-2949<-2383<-577<-1601<-2720<-3225<-1903<-4415<-575<-4116<-4294<-308<-2773<-4800<-4069<-2113<-763<-3845<-314<-965<-2796<-4815<-2521<-3360<-286<-741<-740<-2738<-2093<-297<-296<-4113<-4112<-73<-1852)
Practical Running TIme of Kruskal's: 47ms

==================Graph with Degree of about 20%====================
Using DIJKSTRA(no HEAP): The BW[1850]is 3275 and the path is(1850<-655<-4178<-4376<-3344<-2123<-4689<-1203<-1335<-369<-1109<-1266<-3500<-1770<-3878<-1721<-3233<-2607<-1852)
Practical Running TIme of Dijkstra(No Heap is):177ms

Using HEAP DIJKSTRA :The BW[1850]is 3275 and the path is(1850<-655<-4178<-4376<-3344<-172<-575<-2227<-4239<-2850<-1163<-1507<-1862<-220<-2908<-4620<-3321<-3742<-4041<-2151<-1300<-3701<-4501<-2607<-1852)
Practical Running TIme of Dijkstra(With Heap is):78ms

Using KRUSKAL's: The BW[1850]is 3275 and the path is(1850<-655<-4178<-3362<-761<-1162<-1354<-4231<-1881<-2606<-721<-887<-2633<-3902<-570<-536<-1426<-237<-3392<-1149<-48<-3316<-4416<-2563<-2416<-3239<-2988<-2236<-1300<-3701<-4501<-2607<-1852)
Practical Running TIme of Kruskal's: 375ms
==========

==========

Now Testing using Pair[4] with vertices (s: 2233,t:4667)

=====================Graph with Degree 6 =====================
Using DIJKSTRA(no HEAP): The BW[4667]is 2954 and the path is(4667<-4233<-4492<-3577<-1795<-613<-2131<-2147<-891<-890<-617<-53<-4390<-3746<-2644<-3785<-1942<-270<-1754<-4058<-4871<-4668<-1325<-1092<-1093<-370<-2244<-4149<-4148<-3688<-4194<-2233)
Practical Running TIme of Dijkstra(No Heap is):137ms

Using HEAP DIJKSTRA :The BW[4667]is 2954 and the path is(4667<-4233<-4492<-3577<-1795<-613<-2131<-2147<-891<-302<-1773<-2454<-3985<-4035<-4321<-1358<-4973<-4972<-1244<-2752<-122<-3523<-3527<-1777<-1626<-1754<-2371<-1923<-1022<-1863<-1862<-191<-3978<-400<-1030<-499<-308<-4149<-4148<-3688<-4194<-2233)
Practical Running TIme of Dijkstra(With Heap is):21ms

Using KRUSKAL's: The BW[4667]is 2954 and the path is(4667<-4233<-4492<-3577<-1795<-613<-2131<-2147<-891<-302<-1773<-2454<-3985<-4035<-4321<-1358<-4973<-4972<-1244<-2752<-2177<-2176<-4175<-2658<-1787<-1693<-1025<-1865<-2045<-2143<-4346<-1765<-4929<-1525<-1994<-3781<-134<-830<-426<-1005<-1628<-4650<-1323<-4638<-1488<-3785<-1942<-270<-1754<-4058<-4871<-4668<-1325<-1092<-1093<-370<-2244<-4149<-4148<-3688<-4194<-2233)
Practical Running TIme of Kruskal's: 47ms

==========Graph with Degree of about 20%==========
Using DIJKSTRA(no HEAP): The BW[4667]is 3292 and the path is(4667<-4420<-381<-3879<-437<-288<-1627<-3638<-3905<-2279<-3793<-48<-2711<-2071<-338<-2308<-1856<-1252<-2232<-4880<-172<-1228<-501<-1426<-73<-2520<-2287<-3408<-4861<-359<-2462<-2693<-1554<-2233)
Practical Running TIme of Dijkstra(No Heap is):204ms

Using HEAP DIJKSTRA :The BW[4667]is 3292 and the path is(4667<-4420<-381<-4611<-1872<-382<-945<-1691<-2028<-3480<-4561<-1274<-2325<-2562<-4746<-2376<-2426<-2727<-2357<-563<-73<-2520<-2287<-3408<-4861<-359<-2462<-2693<-1554<-2233)
Practical Running TIme of Dijkstra(With Heap is):60ms

Using KRUSKAL's: The BW[4667]is 3292 and the path is(4667<-4420<-594<-3014<-1037<-3547<-2928<-2713<-3103<-1056<-3533<-437<-288<-1627<-847<-2812<-1578<-3384<-3223<-1851<-1462<-227<-1596<-1223<-2269<-2287<-3408<-4861<-359<-2462<-2693<-1554<-2233)
Practical Running TIme of Kruskal's: 360ms
==========

==========

Now Testing using Pair[5] with vertices (s: 2521,t:2873)

==========Graph with Degree 6 ==========
Using DIJKSTRA(no HEAP): The BW[2873]is 2958 and the path is(2873<-4307<-2462<-950<-1459<-680<-2835<-1455<-3047<-1093<-4912<-3378<-1083<-825<-1583<-2829<-1610<-371<-798<-4083<-3923<-1180<-3067<-2652<-4762<-476<-901<-1560<-1561<-2290<-993<-2120<-1321<-1320<-477<-1316<-1413<-3797<-3279<-2521)
Practical Running TIme of Dijkstra(No Heap is):126ms

Using HEAP DIJKSTRA :The BW[2873]is 2958 and the path is(2873<-4307<-2462<-950<-1459<-680<-2835<-1455<-3047<-1093<-1419<-48<-1441<-1348<-1249<-2357<-4547<-1409<-999<-1686<-712<-226<-575<-2998<-29<-3923<-122<-123<-958<-240<-2102<-1688<-2493<-3370<-553<-89<-3071<-3871<-3093<-2198<-4560<-56<-1656<-3629<-4899<-4273<-3056<-2810<-2811<-2399<-2216<-1199<-1048<-1930<-1931<-1560<-1561<-2290<-993<-2120<-1321<-1320<-477<-1316<-1413<-3797<-3279<-2521)
Practical Running TIme of Dijkstra(With Heap is):25ms

Using KRUSKAL's: The BW[2873]is 2958 and the path is(2873<-4307<-2462<-950<-1459<-680<-2835<-1455<-3047<-1093<-1419<-631<-2701<-1390<-3284<-3639<-882<-508<-300<-4438<-762<-1249<-3262<-1039<-4237<-2204<-2205<-4238<-576<-2771<-1708<-581<-398<-399<-3428<-2581<-3600<-2102<-240<-958<-

123<-122<-3923<-1180<-3067<-2652<-4762<-476<-901<-1560<-1561<-2290<-993<-2120<-1321<-1320<-477<-1316<-1413<-3797<-3279<-2521)
Practical Running TIme of Kruskal's: 44ms

==========Graph with Degree of about 20%==========
Using DIJKSTRA(no HEAP): The BW[2873]is 3296 and the path is(2873<-1330<-4747<-3693<-2272<-4635<-806<-319<-4856<-2054<-2521)
Practical Running TIme of Dijkstra(No Heap is):189ms

Using HEAP DIJKSTRA :The BW[2873]is 3296 and the path is(2873<-1330<-4747<-3693<-2272<-4635<-806<-319<-4856<-2054<-2521)
Practical Running TIme of Dijkstra(With Heap is):52ms

Using KRUSKAL's: The BW[2873]is 3296 and the path is(2873<-3680<-1109<-1806<-3464<-2475<-958<-1194<-4341<-2965<-1959<-4013<-3809<-4536<-949<-1078<-4921<-4641<-1330<-4747<-3693<-2272<-4635<-806<-319<-4856<-2054<-2521)
Practical Running TIme of Kruskal's: 358ms
==========

Now Testing Pair[3] Graph

==========

Now Testing using Pair[1] with vertices (s: 1538,t:4554)

==========Graph with Degree 6 ==========
Using DIJKSTRA(no HEAP): The BW[4554]is 3051 and the path is(4554<-3283<-1770<-1076<-1077<-3087<-2935<-2174<-1602<-1345<-4446<-1016<-2379<-902<-2203<-2063<-4685<-2068<-3448<-3961<-4587<-1615<-1297<-1310<-612<-1784<-3609<-661<-987<-2875<-353<-3117<-3116<-1404<-3277<-528<-650<-1538)
Practical Running TIme of Dijkstra(No Heap is):131ms

Using HEAP DIJKSTRA :The BW[4554]is 3051 and the path is(4554<-3283<-1770<-1076<-1077<-3087<-2935<-2174<-1602<-1345<-4446<-1016<-2379<-1361<-1360<-2521<-526<-829<-2649<-734<-762<-613<-2294<-1093<-1432<-1971<-161<-1626<-1627<-1507<-4903<-1484<-168<-3018<-4214<-556<-4818<-2672<-2676<-4940<-4474<-918<-175<-74<-1852<-1181<-4162<-2701<-475<-1029<-3112<-4826<-1331<-661<-987<-2875<-353<-3117<-3116<-1404<-3277<-528<-650<-1538)
Practical Running TIme of Dijkstra(With Heap is):26ms

Using KRUSKAL's: The BW[4554]is 3051 and the path is(4554<-3283<-1770<-1076<-1077<-3087<-2935<-2174<-1602<-1345<-4446<-1016<-2379<-4800<-3195<-724<-1643<-4152<-4101<-4211<-82<-3735<-4090<-4035<-4063<-2878<-2163<-4636<-2109<-3757<-2695<-4545<-791<-3989<-2288<-1913<-4075<-4074<-4958<-384<-3262<-2741<-2468<-1094<-3568<-1317<-2808<-2716<-2717<-2916<-3794<-1909<-987<-2875<-353<-3117<-3116<-1404<-3277<-528<-650<-1538)
Practical Running TIme of Kruskal's: 43ms

==========Graph with Degree of about 20%==========
Using DIJKSTRA(no HEAP): The BW[4554]is 3281 and the path is(4554<-3677<-1916<-608<-3143<-3142<-2577<-955<-1085<-2579<-4522<-1801<-168<-3011<-214<-4741<-463<-1272<-1287<-667<-233<-1347<-388<-1831<-1609<-672<-2820<-2529<-3239<-2366<-1538)
Practical Running TIme of Dijkstra(No Heap is):181ms

Using HEAP DIJKSTRA :The BW[4554]is 3281 and the path is(4554<-3677<-1916<-413<-1510<-914<-1050<-2272<-3085<-1836<-2025<-3334<-3783<-394<-543<-1472<-129<-805<-4155<-3542<-212<-2366<-1538)
Practical Running TIme of Dijkstra(With Heap is):56ms

Using KRUSKAL's: The BW[4554]is 3281 and the path is(4554<-3677<-1916<-413<-1510<-914<-1050<-2272<-3935<-2983<-3835<-432<-501<-2613<-1906<-2529<-3239<-2366<-1538)
Practical Running TIme of Kruskal's: 345ms
==========

==========

Now Testing using Pair[2] with vertices (s: 3396,t:4851)

==========Graph with Degree 6 ==========
Using DIJKSTRA(no HEAP): The BW[4851]is 2923 and the path is(4851<-3636<-2789<-4700<-3365<-4158<-1211<-1526<-2159<-1617<-572<-783<-4442<-726<-248<-3396)
Practical Running TIme of Dijkstra(No Heap is):137ms

Using HEAP DIJKSTRA :The BW[4851]is 2923 and the path is(4851<-3636<-2789<-4700<-3365<-4158<-1211<-1526<-2159<-1617<-572<-783<-4442<-726<-248<-3396)
Practical Running TIme of Dijkstra(With Heap is):12ms

Using KRUSKAL's: The BW[4851]is 2923 and the path is(4851<-3636<-2789<-4700<-3365<-4158<-1211<-1526<-2159<-1617<-572<-783<-4442<-726<-248<-3396)
Practical Running TIme of Kruskal's: 41ms

==========Graph with Degree of about 20%==========
Using DIJKSTRA(no HEAP): The BW[4851]is 3296 and the path is(4851<-3077<-1499<-1012<-4460<-2707<-100<-4943<-857<-3320<-420<-4069<-2021<-4622<-1658<-1877<-2672<-4608<-4280<-1754<-1477<-302<-505<-4941<-1892<-3326<-186<-1615<-1216<-2542<-4512<-1010<-2509<-4587<-3037<-1819<-2533<-784<-3374<-2169<-705<-2926<-2378<-2058<-205<-4915<-2077<-1255<-3886<-3623<-1333<-2051<-106<-4605<-4138<-3856<-2060<-3396)
Practical Running TIme of Dijkstra(No Heap is):220ms

Using HEAP DIJKSTRA :The BW[4851]is 3296 and the path is(4851<-3077<-1499<-1012<-4460<-2707<-100<-4943<-857<-3320<-420<-4069<-2021<-4622<-1658<-1877<-2672<-4608<-4280<-1754<-1477<-302<-505<-4941<-2403<-4679<-1832<-72<-4028<-2471<-1275<-3224<-3037<-1819<-2533<-784<-3374<-2169<-705<-2926<-2378<-2058<-205<-4915<-2077<-1255<-3886<-3623<-1333<-2051<-106<-4605<-4138<-3856<-2060<-3396)
Practical Running TIme of Dijkstra(With Heap is):67ms

Using KRUSKAL's: The BW[4851]is 3296 and the path is(4851<-3077<-1499<-1012<-4460<-2707<-100<-4943<-857<-3320<-420<-4069<-2021<-4622<-1658<-1877<-2672<-4608<-4280<-1754<-1477<-302<-505<-4941<-2403<-4679<-1832<-72<-4028<-2471<-1275<-3224<-3037<-1819<-2533<-784<-3374<-2169<-705<-2926<-2378<-2058<-205<-4915<-2077<-1255<-3886<-3623<-1333<-2051<-106<-4605<-4138<-3856<-2060<-3396)
Practical Running TIme of Kruskal's: 365ms
==========

==========

Now Testing using Pair[3] with vertices (s: 2015,t:3237)

=====Graph with Degree 6 =====
Using DIJKSTRA(no HEAP): The BW[3237]is 2780 and the path is(3237<-1773<-2740<-445<-153<-4864<-1198<-4920<-4643<-932<-1484<-2701<-3351<-4533<-2014<-2015)
Practical Running TIme of Dijkstra(No Heap is):125ms

Using HEAP DIJKSTRA :The BW[3237]is 2780 and the path is(3237<-1773<-2740<-445<-153<-4864<-2184<-563<-2840<-4742<-219<-3673<-2997<-3396<-1733<-1732<-3351<-4533<-2014<-2015)
Practical Running TIme of Dijkstra(With Heap is):16ms

Using KRUSKAL's: The BW[3237]is 2780 and the path is(3237<-1773<-2740<-2741<-2027<-427<-3980<-1046<-4453<-4365<-4364<-4591<-1878<-1431<-2140<-3040<-3718<-4177<-4796<-1459<-1565<-2500<-3327<-1300<-3925<-871<-1732<-3351<-4533<-2014<-2015)
Practical Running TIme of Kruskal's: 66ms

=====Graph with Degree of about 20%=====
Using DIJKSTRA(no HEAP): The BW[3237]is 3235 and the path is(3237<-169<-246<-2004<-4832<-513<-737<-2475<-2640<-1842<-8<-2068<-171<-3403<-2444<-842<-3148<-452<-3731<-2471<-2015)
Practical Running TIme of Dijkstra(No Heap is):206ms

Using HEAP DIJKSTRA :The BW[3237]is 3235 and the path is(3237<-169<-2010<-2582<-3947<-1314<-1278<-4860<-3323<-966<-579<-68<-4361<-1909<-2642<-4137<-4429<-1915<-1695<-4322<-525<-4098<-3529<-2368<-1198<-206<-2328<-2101<-1211<-287<-2193<-863<-4031<-1495<-1598<-2233<-2062<-32<-2337<-115<-4377<-2933<-644<-920<-4106<-3212<-4173<-2628<-2733<-4744<-1842<-8<-2068<-171<-3403<-2444<-842<-3148<-452<-3731<-2471<-2015)
Practical Running TIme of Dijkstra(With Heap is):76ms

Using KRUSKAL's: The BW[3237]is 3235 and the path is(3237<-169<-246<-2004<-4832<-513<-737<-2475<-2640<-1842<-8<-2068<-261<-967<-171<-3403<-2444<-842<-3148<-452<-3731<-2471<-2015)
Practical Running TIme of Kruskal's: 349ms
==========

==========

Now Testing using Pair[4] with vertices (s: 2467,t:1010)

=====Graph with Degree 6 =====
Using DIJKSTRA(no HEAP): The BW[1010]is 2978 and the path is(1010<-3739<-4632<-1339<-3165<-3821<-2671<-1840<-2551<-3776<-4804<-150<-4049<-2224<-2225<-4348<-2467)
Practical Running TIme of Dijkstra(No Heap is):126ms

Using HEAP DIJKSTRA :The BW[1010]is 2978 and the path is(1010<-2620<-4640<-3349<-3348<-4199<-4366<-3562<-793<-3937<-2691<-1488<-385<-4408<-4829<-1307<-1306<-2233<-4960<-719<-1575<-4901<-562<-405<-2159<-4272<-4133<-4059<-2618<-1521<-3318<-202<-1087<-2550<-2551<-3776<-4804<-150<-4049<-2224<-2225<-4348<-2467)
Practical Running TIme of Dijkstra(With Heap is):27ms

Using KRUSKAL's: The BW[1010]is 2978 and the path is(1010<-3739<-4632<-1339<-3165<-3821<-2671<-1840<-2551<-3776<-4804<-150<-4049<-2224<-2225<-4348<-2467)
Practical Running TIme of Kruskal's: 69ms

==========Graph with Degree of about 20%==========
Using DIJKSTRA(no HEAP): The BW[1010]is 3256 and the path is(1010<-4051<-2530<-4945<-2205<-52<-53<-3055<-958<-2624<-4180<-1367<-3034<-4266<-1548<-4801<-374<-686<-901<-889<-1634<-519<-1538<-4452<-848<-1382<-1760<-4619<-2006<-2203<-4097<-3493<-4089<-4088<-1502<-1355<-2964<-4288<-4011<-2467)
Practical Running TIme of Dijkstra(No Heap is):187ms

Using HEAP DIJKSTRA :The BW[1010]is 3256 and the path is(1010<-4051<-2530<-4945<-2205<-52<-53<-3055<-958<-4150<-109<-2868<-1141<-2840<-179<-138<-579<-2293<-1154<-127<-2268<-37<-1467<-522<-724<-4097<-3493<-4089<-4088<-1502<-1355<-2964<-4288<-4011<-2467)
Practical Running TIme of Dijkstra(With Heap is):60ms

Using KRUSKAL's: The BW[1010]is 3256 and the path is(1010<-4051<-2530<-4945<-2205<-52<-53<-3055<-958<-4150<-109<-2868<-1141<-1740<-1766<-874<-2239<-4874<-1666<-80<-2631<-1062<-1267<-2715<-1393<-3634<-1832<-2251<-3864<-1605<-3062<-305<-107<-2007<-4538<-3282<-4829<-1574<-4661<-4769<-51<-159<-1978<-2624<-4180<-626<-523<-3674<-1500<-711<-97<-3194<-895<-1901<-2590<-142<-2418<-341<-1<-937<-1539<-3326<-1624<-939<-2708<-4941<-1868<-1686<-3742<-3945<-2030<-1882<-4029<-2621<-3256<-164<-4086<-4097<-3493<-4089<-4088<-1502<-1355<-2964<-4288<-4011<-2467)
Practical Running TIme of Kruskal's: 398ms
==========

==========

Now Testing using Pair[5] with vertices (s: 2084,t:3842)

==========Graph with Degree 6 ==========
Using DIJKSTRA(no HEAP): The BW[3842]is 3030 and the path is(3842<-2874<-2745<-4068<-2132<-1052<-1336<-80<-4253<-4658<-3750<-3635<-2768<-2978<-2323<-840<-670<-2084)
Practical Running TIme of Dijkstra(No Heap is):126ms

Using HEAP DIJKSTRA :The BW[3842]is 3030 and the path is(3842<-2874<-2745<-4068<-2132<-1052<-1336<-80<-4253<-4658<-3750<-3635<-2768<-2978<-2323<-840<-670<-2084)
Practical Running TIme of Dijkstra(With Heap is):12ms

Using KRUSKAL's: The BW[3842]is 3030 and the path is(3842<-2874<-2745<-4068<-2132<-1052<-1336<-80<-4253<-4658<-3750<-3635<-2768<-2978<-2323<-840<-670<-2084)
Practical Running TIme of Kruskal's: 51ms

==========Graph with Degree of about 20%==========
Using DIJKSTRA(no HEAP): The BW[3842]is 3286 and the path is(3842<-1877<-1138<-861<-3129<-273<-3715<-156<-2672<-4805<-1237<-4612<-2320<-3952<-2084)
Practical Running TIme of Dijkstra(No Heap is):175ms

Using HEAP DIJKSTRA :The BW[3842]is 3286 and the path is(3842<-1877<-1138<-4849<-3684<-2426<-4727<-1377<-4571<-4349<-4925<-4924<-1857<-4781<-350<-2436<-106<-2320<-3952<-2084)
Practical Running TIme of Dijkstra(With Heap is):59ms

Using KRUSKAL's: The BW[3842]is 3286 and the path is(3842<-1877<-1138<-1956<-643<-851<-51<-732<-1870<-2112<-314<-1085<-4998<-4179<-2468<-709<-1616<-890<-2320<-3952<-2084)
Practical Running TIme of Kruskal's: 358ms
==========

Now Testing Pair[4] Graph

==========

Now Testing using Pair[1] with vertices (s: 589,t:3278)

==========Graph with Degree 6 ==========
Using DIJKSTRA(no HEAP): The BW[3278]is 2882 and the path is(3278<-1087<-2575<-3407<-4934<-2412<-2311<-1110<-589)
Practical Running TIme of Dijkstra(No Heap is):123ms

Using HEAP DIJKSTRA :The BW[3278]is 2882 and the path is(3278<-1087<-4851<-1261<-1544<-4332<-3301<-4183<-4530<-3980<-1992<-1993<-2197<-1269<-302<-4677<-4676<-1584<-87<-4093<-4152<-3709<-3798<-2311<-1110<-589)
Practical Running TIme of Dijkstra(With Heap is):14ms

Using KRUSKAL's: The BW[3278]is 2882 and the path is(3278<-4852<-4950<-175<-3827<-4376<-4377<-1006<-1569<-2350<-4447<-159<-1955<-1954<-4319<-2343<-2509<-352<-955<-720<-535<-1110<-589)
Practical Running TIme of Kruskal's: 33ms

==========Graph with Degree of about 20%==========
Using DIJKSTRA(no HEAP): The BW[3278]is 3297 and the path is(3278<-44<-791<-4506<-814<-1810<-4295<-734<-2699<-4882<-4719<-4043<-1057<-4074<-41<-801<-2018<-1167<-4204<-4014<-3526<-567<-2157<-1047<-1724<-2059<-2675<-2859<-1656<-3530<-2246<-4909<-4631<-880<-4501<-2351<-3996<-4614<-500<-1973<-4102<-1640<-3298<-18<-1837<-1951<-1672<-1607<-1659<-589)
Practical Running TIme of Dijkstra(No Heap is):183ms

Using HEAP DIJKSTRA :The BW[3278]is 3297 and the path is(3278<-44<-791<-4506<-814<-1810<-570<-4677<-832<-1182<-1025<-3291<-2520<-1804<-4006<-1496<-1253<-1007<-377<-4159<-4102<-1640<-3298<-18<-1837<-1951<-1672<-1607<-1659<-589)
Practical Running TIme of Dijkstra(With Heap is):55ms

Using KRUSKAL's: The BW[3278]is 3297 and the path is(3278<-44<-791<-1051<-2998<-1047<-2157<-36<-384<-2908<-2290<-378<-1761<-2562<-833<-3072<-476<-2043<-774<-1011<-3216<-2745<-4614<-3996<-2351<-2345<-2407<-2205<-2322<-4159<-4102<-1640<-3298<-18<-1837<-1951<-1672<-1607<-1659<-589)
Practical Running TIme of Kruskal's: 349ms
==========

==========

Now Testing using Pair[2] with vertices (s: 1437,t:438)

==========Graph with Degree 6 ==========

Using DIJKSTRA(no HEAP): The BW[438]is 3038 and the path is(438<-1288<-685<-707<-4760<-4761<-3339<-1115<-4880<-2438<-1144<-2603<-2602<-1611<-2222<-2964<-2600<-2344<-1839<-1313<-4247<-4246<-4804<-4138<-4260<-1437)
Practical Running TIme of Dijkstra(No Heap is):134ms

Using HEAP DIJKSTRA :The BW[438]is 3038 and the path is(438<-1288<-685<-707<-4760<-550<-551<-3446<-4619<-4602<-4603<-3298<-849<-4470<-1864<-4115<-2652<-1075<-2013<-1286<-1501<-1668<-791<-525<-524<-868<-4820<-4062<-4977<-4727<-1976<-704<-1648<-2603<-2602<-1611<-2222<-2964<-2600<-2344<-1839<-1313<-4247<-4246<-4804<-4138<-4260<-1437)
Practical Running TIme of Dijkstra(With Heap is):23ms

Using KRUSKAL's: The BW[438]is 3038 and the path is(438<-1052<-2747<-3117<-180<-750<-4850<-3487<-2369<-4954<-111<-968<-4639<-2395<-4386<-1293<-3342<-2469<-2632<-993<-265<-2721<-1236<-1237<-2024<-2661<-2660<-1605<-1370<-4068<-2228<-2229<-2761<-687<-3154<-792<-2600<-2344<-1839<-1313<-4247<-4246<-4804<-4138<-4260<-1437)
Practical Running TIme of Kruskal's: 39ms

==========Graph with Degree of about 20%==========
Using DIJKSTRA(no HEAP): The BW[438]is 3300 and the path is(438<-687<-436<-4632<-1970<-143<-2061<-3151<-1559<-961<-3793<-1781<-1752<-1654<-1100<-2479<-4813<-1437)
Practical Running TIme of Dijkstra(No Heap is):176ms

Using HEAP DIJKSTRA :The BW[438]is 3300 and the path is(438<-687<-436<-4632<-4641<-479<-4364<-2543<-2886<-73<-4707<-1592<-968<-1363<-3320<-961<-3793<-1781<-1752<-1654<-1100<-2479<-4813<-1437)
Practical Running TIme of Dijkstra(With Heap is):56ms

Using KRUSKAL's: The BW[438]is 3300 and the path is(438<-687<-436<-1515<-1514<-2820<-618<-276<-2096<-1592<-968<-1363<-3320<-961<-3793<-1781<-1752<-1654<-1100<-2479<-4813<-1437)
Practical Running TIme of Kruskal's: 352ms
==========

==========

Now Testing using Pair[3] with vertices (s: 1539,t:563)

==========Graph with Degree 6 ==========
Using DIJKSTRA(no HEAP): The BW[563]is 3012 and the path is(563<-3074<-4945<-1824<-42<-4406<-1474<-247<-4324<-4345<-690<-347<-1965<-1150<-539<-328<-329<-4547<-4896<-4446<-76<-92<-93<-1228<-1435<-1266<-1267<-4073<-2804<-387<-4302<-4303<-2579<-1064<-95<-2465<-3985<-1539)
Practical Running TIme of Dijkstra(No Heap is):129ms

Using HEAP DIJKSTRA :The BW[563]is 3012 and the path is(563<-3074<-4945<-1824<-42<-4406<-4173<-4172<-3994<-912<-150<-3158<-2380<-2381<-415<-2888<-4082<-4683<-1765<-86<-2029<-531<-2661<-2493<-1661<-2442<-3765<-4266<-2724<-1888<-217<-2379<-270<-69<-4276<-2983<-4250<-608<-1126<-506<-3764<-1039<-1578<-4908<-4791<-3262<-4974<-866<-2569<-1293<-2470<-3871<-2683<-3864<-1594<-1655<-95<-2465<-3985<-1539)
Practical Running TIme of Dijkstra(With Heap is):25ms

Using KRUSKAL's: The BW[563]is 3012 and the path is(563<-3074<-4945<-1824<-42<-4406<-4173<-4172<-3994<-912<-150<-3158<-2380<-2381<-415<-2888<-4082<-3579<-1822<-4151<-2393<-1463<-4889<-2030<-4783<-2518<-3670<-3583<-3721<-3642<-309<-4674<-90<-19<-2663<-2662<-2594<-3090<-1967<-

2234<-1645<-3091<-3330<-4597<-4581<-3128<-2341<-3856<-4817<-2818<-3609<-3661<-1283<-4584<-512<-1440<-198<-172<-3490<-1520<-2630<-4509<-4147<-1887<-1665<-2774<-2204<-4215<-3555<-2975<-1988<-1276<-307<-1506<-233<-4191<-195<-2073<-51<-1639<-872<-329<-4547<-4896<-4446<-76<-92<-93<-1228<-1435<-1266<-1267<-4073<-2804<-387<-4302<-4303<-2579<-1064<-95<-2465<-3985<-1539)
Practical Running TIme of Kruskal's: 156ms

==========Graph with Degree of about 20%==========
Using DIJKSTRA(no HEAP): The BW[563]is 3201 and the path is(563<-1135<-2972<-4101<-1091<-1385<-3130<-2572<-1670<-369<-3558<-2526<-4362<-4807<-857<-1545<-2190<-1798<-1451<-474<-1024<-1087<-701<-771<-1248<-1253<-2527<-136<-305<-199<-606<-1365<-4411<-3997<-4345<-3431<-2377<-1412<-4320<-3245<-807<-1781<-4513<-3204<-422<-4005<-290<-3052<-1539)
Practical Running TIme of Dijkstra(No Heap is):215ms

Using HEAP DIJKSTRA :The BW[563]is 3201 and the path is(563<-1135<-2972<-4101<-1091<-1385<-3130<-2572<-1670<-369<-3558<-2526<-4362<-4807<-857<-1545<-2190<-1798<-1451<-2983<-4137<-2255<-3353<-283<-3644<-2619<-2726<-2435<-1698<-901<-2835<-4608<-1783<-3308<-3245<-807<-1781<-4513<-3204<-422<-4005<-290<-3052<-1539)
Practical Running TIme of Dijkstra(With Heap is):63ms

Using KRUSKAL's: The BW[563]is 3201 and the path is(563<-1135<-479<-1706<-4887<-2496<-2003<-561<-2456<-3193<-326<-939<-2292<-3593<-2995<-2320<-55<-2864<-2647<-4027<-3365<-4161<-1621<-3803<-627<-3693<-824<-372<-3100<-3892<-3684<-4522<-120<-529<-1260<-2885<-2272<-638<-1066<-4115<-4815<-1973<-3450<-3433<-4265<-3618<-930<-283<-3644<-2619<-2726<-2435<-1698<-901<-2835<-4608<-1783<-3308<-3245<-807<-1781<-4513<-3204<-422<-4005<-290<-3052<-1539)
Practical Running TIme of Kruskal's: 359ms
==========

==========

Now Testing using Pair[4] with vertices (s: 3268,t:3026)

==========Graph with Degree 6 ==========
Using DIJKSTRA(no HEAP): The BW[3026]is 2995 and the path is(3026<-3672<-4319<-1305<-4970<-3596<-4185<-405<-3619<-2341<-2340<-4607<-2572<-2786<-3716<-1015<-3278<-4088<-3428<-3490<-3483<-2096<-893<-1225<-1788<-4187<-4186<-3302<-2519<-4518<-2609<-2685<-4423<-3524<-17<-3303<-2599<-4990<-792<-783<-3659<-1654<-2742<-4854<-2913<-2134<-1501<-2608<-3268)
Practical Running TIme of Dijkstra(No Heap is):131ms

Using HEAP DIJKSTRA :The BW[3026]is 2995 and the path is(3026<-3672<-4319<-1305<-4970<-3596<-4185<-405<-3619<-2341<-2340<-4607<-2572<-2786<-3716<-4123<-4122<-4891<-242<-1181<-2879<-1820<-2710<-2711<-4998<-2811<-4220<-3172<-2014<-1155<-2125<-2846<-533<-1267<-3549<-976<-3740<-2774<-2626<-4405<-2962<-2409<-2134<-1501<-2608<-3268)
Practical Running TIme of Dijkstra(With Heap is):27ms

Using KRUSKAL's: The BW[3026]is 2995 and the path is(3026<-3672<-4319<-1305<-4970<-3596<-4185<-3125<-3059<-2964<-4820<-4049<-489<-658<-3554<-3751<-163<-2581<-2031<-4881<-3318<-1776<-4855<-1535<-3003<-4004<-4558<-1499<-4612<-3241<-3240<-742<-741<-1341<-230<-4929<-3022<-99<-98<-3648<-467<-134<-4876<-329<-2530<-3395<-1267<-3549<-976<-3740<-2774<-2626<-4405<-2962<-2409<-2134<-1501<-2608<-3268)
Practical Running TIme of Kruskal's: 52ms

==========Graph with Degree of about 20%==========
Using DIJKSTRA(no HEAP): The BW[3026]is 3306 and the path is(3026<-4046<-2616<-118<-1009<-1609<-4396<-4121<-4582<-2102<-4010<-3327<-662<-3224<-3872<-634<-1913<-528<-1631<-2176<-2515<-1885<-1663<-3121<-295<-4848<-1070<-1260<-4639<-1156<-3235<-2193<-2485<-4179<-706<-1586<-4269<-3268)
Practical Running TIme of Dijkstra(No Heap is):181ms

Using HEAP DIJKSTRA :The BW[3026]is 3306 and the path is(3026<-4046<-2616<-118<-1009<-1609<-4396<-4121<-4582<-2102<-4010<-3327<-662<-3224<-3872<-634<-1913<-528<-1631<-2176<-2515<-1885<-1663<-3121<-295<-4848<-1070<-1260<-4639<-1156<-3235<-2193<-2485<-4179<-706<-1586<-4269<-3268)
Practical Running TIme of Dijkstra(With Heap is):61ms

Using KRUSKAL's: The BW[3026]is 3306 and the path is(3026<-4046<-2616<-118<-1009<-1609<-4396<-4121<-4582<-2102<-4010<-3327<-662<-3224<-3872<-634<-1913<-528<-1631<-2176<-2515<-1885<-1663<-3121<-295<-4848<-1070<-1260<-4639<-1156<-3235<-2193<-2485<-4179<-706<-1586<-4269<-3268)
Practical Running TIme of Kruskal's: 358ms
==========

==========

Now Testing using Pair[5] with vertices (s: 1025,t:2993)

==========Graph with Degree 6 ==========
Using DIJKSTRA(no HEAP): The BW[2993]is 2346 and the path is(2993<-2992<-1343<-1947<-2995<-4823<-2974<-1025)
Practical Running TIme of Dijkstra(No Heap is):124ms

Using HEAP DIJKSTRA :The BW[2993]is 2346 and the path is(2993<-2992<-1383<-3777<-3776<-4845<-4022<-1487<-656<-254<-255<-4069<-2185<-1846<-2603<-702<-2974<-1025)
Practical Running TIme of Dijkstra(With Heap is):11ms

Using KRUSKAL's: The BW[2993]is 2346 and the path is(2993<-2992<-3004<-315<-4962<-1378<-989<-2254<-1466<-1407<-2050<-4369<-4368<-4276<-608<-3946<-3006<-4180<-4596<-900<-2414<-2530<-2531<-4618<-1058<-3697<-4436<-395<-4107<-3412<-3413<-4635<-1553<-4961<-451<-450<-997<-739<-371<-2381<-3936<-2058<-4499<-3241<-2826<-4641<-796<-2431<-1450<-4153<-2974<-1025)
Practical Running TIme of Kruskal's: 40ms

==========Graph with Degree of about 20%==========
Using DIJKSTRA(no HEAP): The BW[2993]is 3270 and the path is(2993<-3137<-3599<-63<-411<-77<-2795<-3303<-4162<-3<-3324<-1033<-3430<-4435<-1167<-1366<-1617<-962<-3714<-4360<-3408<-1258<-2671<-3165<-2645<-2054<-3131<-1866<-2256<-1526<-1664<-1027<-1428<-291<-4748<-4120<-1025)
Practical Running TIme of Dijkstra(No Heap is):196ms

Using HEAP DIJKSTRA :The BW[2993]is 3270 and the path is(2993<-151<-1674<-1369<-1083<-322<-2379<-4529<-2077<-947<-593<-4163<-4339<-613<-2349<-4517<-1350<-2973<-3569<-4647<-1847<-673<-765<-3127<-1969<-4686<-1338<-2225<-3604<-2709<-3<-4162<-1426<-102<-1114<-3087<-974<-1575<-511<-3189<-3140<-578<-3365<-4674<-3233<-4028<-1309<-135<-1201<-3340<-11<-3317<-3881<-1745<-4803<-291<-4748<-4120<-1025)
Practical Running TIme of Dijkstra(With Heap is):101ms

Using KRUSKAL's: The BW[2993]is 3270 and the path is(2993<-151<-1674<-1369<-1083<-1082<-1149<-1789<-3156<-1136<-576<-3589<-1907<-962<-1617<-1742<-1478<-1971<-343<-2307<-921<-4446<-3406<-4547<-2597<-2675<-2688<-2714<-233<-391<-410<-2380<-2509<-73<-915<-203<-202<-721<-2199<-

1075<-3971<-2130<-2<-1285<-371<-4102<-3348<-482<-1769<-1931<-4452<-2251<-1664<-1027<-1428<-291<-4748<-4120<-1025)
Practical Running TIme of Kruskal's: 390ms
==========

Now Testing Pair[5] Graph

==========

Now Testing using Pair[1] with vertices (s: 1192,t:2520)

==========Graph with Degree 6 ==========
Using DIJKSTRA(no HEAP): The BW[2520]is 2957 and the path is(2520<-2521<-2138<-3164<-3850<-3851<-4450<-1481<-2739<-1020<-979<-2359<-4002<-4805<-4207<-2509<-4393<-2652<-3567<-2096<-2943<-1856<-2766<-4518<-2793<-4389<-1275<-2733<-3151<-382<-4237<-4224<-2443<-2442<-405<-1169<-1193<-1192)
Practical Running TIme of Dijkstra(No Heap is):132ms

Using HEAP DIJKSTRA :The BW[2520]is 2957 and the path is(2520<-2521<-2138<-3164<-3850<-3851<-1239<-83<-4072<-767<-3144<-2409<-760<-72<-1620<-1260<-2443<-2442<-405<-1169<-1193<-1192)
Practical Running TIme of Dijkstra(With Heap is):18ms

Using KRUSKAL's: The BW[2520]is 2957 and the path is(2520<-2521<-2138<-3801<-3618<-2587<-3120<-4100<-1816<-1552<-3667<-30<-3545<-3544<-4194<-2121<-2847<-1962<-4512<-3233<-1564<-341<-697<-2412<-4550<-4637<-4636<-2526<-295<-3578<-3399<-401<-4350<-488<-489<-3876<-3978<-2875<-3518<-3275<-382<-4237<-4224<-2443<-2442<-405<-1169<-1193<-1192)
Practical Running TIme of Kruskal's: 73ms

==========Graph with Degree of about 20%==========
Using DIJKSTRA(no HEAP): The BW[2520]is 3286 and the path is(2520<-2684<-2683<-912<-890<-1823<-2061<-1482<-478<-1184<-174<-1535<-769<-1759<-3044<-1741<-1521<-483<-2232<-612<-146<-2196<-2508<-112<-1933<-1947<-3392<-2192<-2446<-3565<-664<-4636<-2701<-3927<-1192)
Practical Running TIme of Dijkstra(No Heap is):216ms

Using HEAP DIJKSTRA :The BW[2520]is 3286 and the path is(2520<-2684<-2683<-912<-890<-2419<-258<-736<-1036<-110<-1740<-1809<-2796<-1470<-1747<-2738<-1819<-1818<-613<-569<-805<-425<-471<-1335<-3443<-2278<-786<-1995<-101<-81<-2130<-4902<-3146<-2564<-2283<-4117<-2282<-3927<-1192)
Practical Running TIme of Dijkstra(With Heap is):97ms

Using KRUSKAL's: The BW[2520]is 3286 and the path is(2520<-2684<-2683<-3751<-2145<-2726<-1910<-2714<-2427<-4351<-1554<-1532<-737<-4513<-4171<-4399<-3644<-4813<-2822<-1074<-3928<-1364<-35<-4804<-1089<-1695<-4543<-183<-4608<-1594<-832<-4470<-3565<-664<-4636<-2701<-3927<-1192)
Practical Running TIme of Kruskal's: 393ms
==========

==========

Now Testing using Pair[2] with vertices (s: 4542,t:732)

==========Graph with Degree 6 ==========

Using DIJKSTRA(no HEAP): The BW[732]is 2835 and the path is(732<-1889<-1047<-2377<-3674<-2191<-4583<-2262<-3141<-2508<-2212<-3650<-944<-2519<-2880<-464<-3063<-3766<-1265<-4081<-3383<-2665<-3898<-4900<-4542)
Practical Running TIme of Dijkstra(No Heap is):128ms

Using HEAP DIJKSTRA :The BW[732]is 2835 and the path is(732<-1889<-1047<-2377<-3674<-2191<-4583<-2262<-3141<-2508<-2212<-3650<-944<-945<-4835<-4547<-3480<-2567<-2566<-2131<-4336<-4081<-3383<-2665<-3898<-4900<-4542)
Practical Running TIme of Dijkstra(With Heap is):20ms

Using KRUSKAL's: The BW[732]is 2835 and the path is(732<-1889<-1047<-2377<-3674<-2191<-2236<-1566<-2425<-3328<-4452<-3906<-705<-704<-1686<-3609<-4437<-2531<-2605<-2227<-290<-3596<-2391<-2118<-4391<-4880<-3161<-3041<-411<-4072<-2051<-3329<-1727<-1225<-1399<-1164<-1151<-3900<-3692<-2873<-232<-1729<-4970<-3919<-3076<-3089<-980<-4286<-4529<-34<-4789<-1341<-1404<-568<-3063<-3766<-1265<-4081<-3383<-2665<-3898<-4900<-4542)
Practical Running TIme of Kruskal's: 80ms

==========Graph with Degree of about 20%==========
Using DIJKSTRA(no HEAP): The BW[732]is 3206 and the path is(732<-2837<-442<-1371<-772<-4542)
Practical Running TIme of Dijkstra(No Heap is):213ms

Using HEAP DIJKSTRA :The BW[732]is 3206 and the path is(732<-2837<-4928<-3607<-3720<-1943<-2416<-3205<-3420<-834<-2705<-59<-1145<-4050<-4786<-473<-381<-229<-361<-1927<-2<-286<-4790<-3777<-2969<-570<-1026<-3427<-162<-2211<-614<-2935<-4407<-34<-2175<-2929<-1327<-3615<-1646<-64<-2091<-3323<-1371<-772<-4542)
Practical Running TIme of Dijkstra(With Heap is):71ms

Using KRUSKAL's: The BW[732]is 3206 and the path is(732<-2837<-4928<-1888<-1088<-350<-1934<-531<-4799<-4833<-1487<-3086<-1550<-1622<-2257<-1470<-4355<-179<-2114<-4689<-468<-3066<-1970<-1216<-838<-396<-2516<-4824<-829<-3830<-3095<-72<-73<-2296<-4036<-154<-3450<-3539<-458<-2888<-4781<-1490<-386<-1625<-1953<-2966<-4319<-450<-867<-1442<-4575<-853<-4114<-463<-474<-3050<-3534<-772<-4542)
Practical Running TIme of Kruskal's: 370ms
==========

==========

Now Testing using Pair[3] with vertices (s: 4095,t:4939)

==========Graph with Degree 6 ==========
Using DIJKSTRA(no HEAP): The BW[4939]is 2863 and the path is(4939<-1178<-3407<-2195<-1158<-1922<-259<-4440<-4031<-2695<-3629<-4624<-1958<-4408<-4802<-2701<-1566<-4095)
Practical Running TIme of Dijkstra(No Heap is):124ms

Using HEAP DIJKSTRA :The BW[4939]is 2863 and the path is(4939<-1178<-3407<-2195<-1158<-1392<-819<-741<-3937<-752<-2425<-286<-842<-3869<-2984<-2730<-282<-372<-2209<-1155<-2827<-913<-4337<-307<-3126<-148<-3540<-1162<-1163<-1098<-1099<-3858<-487<-216<-4796<-1864<-2701<-1566<-4095)
Practical Running TIme of Dijkstra(With Heap is):34ms

Using KRUSKAL's: The BW[4939]is 2863 and the path is(4939<-1178<-3407<-2195<-1158<-2822<-4254<-4761<-4760<-3530<-924<-101<-3084<-3052<-1491<-304<-361<-79<-3516<-3967<-4247<-3566<-2992<-

3880<-3198<-667<-1644<-164<-3937<-752<-2425<-286<-842<-3869<-2984<-1523<-2488<-4955<-4841<-2031<-893<-1758<-1958<-4408<-4802<-2701<-1566<-4095)
Practical Running TIme of Kruskal's: 72ms

==========Graph with Degree of about 20%==========
Using DIJKSTRA(no HEAP): The BW[4939]is 3258 and the path is(4939<-377<-2896<-455<-1282<-3434<-1577<-3161<-3582<-1250<-4590<-3071<-3866<-2773<-1387<-2634<-1857<-4095)
Practical Running TIme of Dijkstra(No Heap is):208ms

Using HEAP DIJKSTRA :The BW[4939]is 3258 and the path is(4939<-377<-2896<-455<-3171<-595<-729<-3562<-468<-4081<-358<-1242<-596<-1914<-629<-897<-3273<-2188<-42<-70<-1177<-390<-1645<-2237<-4080<-4590<-3071<-3866<-2773<-1387<-2634<-1857<-4095)
Practical Running TIme of Dijkstra(With Heap is):62ms

Using KRUSKAL's: The BW[4939]is 3258 and the path is(4939<-377<-2896<-455<-1153<-950<-321<-672<-606<-2760<-2950<-3196<-2097<-3157<-2152<-2481<-2595<-119<-2041<-466<-185<-4330<-2092<-4220<-530<-666<-2735<-2634<-1857<-4095)
Practical Running TIme of Kruskal's: 354ms
==========

==========

Now Testing using Pair[4] with vertices (s: 1984,t:3430)

==========Graph with Degree 6 ==========
Using DIJKSTRA(no HEAP): The BW[3430]is 2921 and the path is(3430<-4850<-403<-4181<-2142<-3176<-3008<-3009<-932<-3229<-4377<-1239<-2288<-3091<-3090<-3805<-2584<-2893<-519<-3002<-4809<-1984)
Practical Running TIme of Dijkstra(No Heap is):129ms

Using HEAP DIJKSTRA :The BW[3430]is 2921 and the path is(3430<-4850<-403<-4181<-2142<-3176<-3008<-3009<-932<-3229<-4377<-1239<-2288<-3091<-3090<-3805<-2584<-2893<-519<-3002<-4809<-1984)
Practical Running TIme of Dijkstra(With Heap is):15ms

Using KRUSKAL's: The BW[3430]is 2921 and the path is(3430<-4850<-403<-4181<-2142<-3176<-3008<-3009<-932<-3229<-4377<-3364<-4482<-3239<-3870<-4196<-2617<-4238<-902<-2584<-2893<-519<-3002<-4809<-1984)
Practical Running TIme of Kruskal's: 36ms

==========Graph with Degree of about 20%==========
Using DIJKSTRA(no HEAP): The BW[3430]is 3290 and the path is(3430<-4386<-390<-3914<-677<-823<-616<-4483<-3934<-4695<-1121<-567<-939<-304<-1077<-2173<-4339<-1706<-79<-1147<-2193<-3719<-2301<-1679<-2080<-841<-751<-2515<-1166<-428<-3254<-1800<-1545<-1957<-2634<-1657<-4263<-1372<-3487<-1850<-1984)
Practical Running TIme of Dijkstra(No Heap is):220ms

Using HEAP DIJKSTRA :The BW[3430]is 3290 and the path is(3430<-4386<-390<-3914<-677<-3291<-2071<-4471<-4470<-3586<-231<-2909<-1011<-1271<-1604<-4283<-21<-1837<-2766<-3232<-2321<-3566<-2926<-1850<-1984)
Practical Running TIme of Dijkstra(With Heap is):55ms

Using KRUSKAL's: The BW[3430]is 3290 and the path is(3430<-4386<-390<-3914<-677<-823<-616<-138<-4297<-4573<-2079<-2435<-2921<-1125<-673<-1245<-3566<-2926<-1850<-1984)
Practical Running TIme of Kruskal's: 352ms
==========

==========

Now Testing using Pair[5] with vertices (s: 174,t:4584)

==========Graph with Degree 6 ==========
Using DIJKSTRA(no HEAP): The BW[4584]is 2697 and the path is(4584<-173<-172<-2627<-1737<-1097<-2467<-3102<-3103<-1786<-1064<-2562<-4168<-1503<-3635<-174)
Practical Running TIme of Dijkstra(No Heap is):125ms

Using HEAP DIJKSTRA :The BW[4584]is 2697 and the path is(4584<-4419<-4856<-2189<-4300<-2100<-29<-2650<-2313<-2806<-4193<-3389<-2773<-3690<-1250<-4998<-4732<-3677<-1481<-2768<-2654<-989<-159<-2705<-2704<-1051<-117<-3640<-3681<-3680<-1531<-4704<-738<-2563<-2562<-4168<-1503<-3635<-174)
Practical Running TIme of Dijkstra(With Heap is):20ms

Using KRUSKAL's: The BW[4584]is 2697 and the path is(4584<-4419<-1030<-4579<-1592<-4119<-606<-2356<-2274<-1192<-4243<-1598<-2400<-4488<-3843<-3100<-1291<-1290<-783<-378<-796<-1591<-4889<-2581<-2685<-1781<-3891<-1869<-3165<-128<-1800<-1801<-3850<-126<-3530<-775<-3936<-110<-4989<-4493<-3527<-1450<-1068<-3405<-4548<-4256<-1643<-3075<-1489<-918<-2877<-2219<-4452<-4316<-2541<-1226<-3305<-1844<-872<-873<-3173<-1031<-2200<-1978<-2094<-1699<-2204<-4530<-2080<-3438<-377<-608<-164<-165<-2582<-3116<-1122<-1225<-2483<-2437<-300<-3092<-4869<-4211<-2615<-2689<-658<-4764<-2405<-1443<-627<-3069<-2264<-2562<-4168<-1503<-3635<-174)
Practical Running TIme of Kruskal's: 57ms

==========Graph with Degree of about 20%==========
Using DIJKSTRA(no HEAP): The BW[4584]is 3276 and the path is(4584<-1646<-15<-869<-1928<-1299<-412<-2196<-4223<-174)
Practical Running TIme of Dijkstra(No Heap is):176ms

Using HEAP DIJKSTRA :The BW[4584]is 3276 and the path is(4584<-1646<-15<-869<-1128<-4447<-4401<-3858<-3923<-2772<-1440<-4186<-1338<-4585<-3883<-297<-2831<-201<-3393<-537<-4228<-2178<-412<-2196<-4223<-174)
Practical Running TIme of Dijkstra(With Heap is):56ms

Using KRUSKAL's: The BW[4584]is 3276 and the path is(4584<-1646<-2602<-2477<-3063<-2451<-1935<-856<-3464<-4493<-3611<-18<-1665<-576<-1764<-989<-73<-292<-245<-3162<-1266<-4056<-2840<-4399<-1433<-3806<-4958<-1427<-3453<-1615<-1086<-351<-4912<-3914<-2948<-3408<-2930<-3942<-4921<-1386<-3281<-473<-2490<-2967<-1739<-1419<-308<-1673<-4470<-3696<-307<-2810<-3851<-2513<-1095<-1162<-1783<-1740<-1463<-1234<-2606<-2196<-4223<-174)
Practical Running TIme of Kruskal's: 353ms
==========

A model Adjacency list of a graph can be found at graph.txt