Report on runtimes of BFS on Adjacency List implementation of Graph:

For 1000 vertices:

|  |  |
| --- | --- |
| Edges | Time (micro sec) |
| 1000 | 0 |
| 2000 | 99 |
| 4000 | 398 |
| 8000 | 1196 |
| 16000 | 3695 |
| 32000 | 12170 |
| 64000 | 39893 |

For 2000 vertices:

|  |  |
| --- | --- |
| Edges | Time(micro sec) |
| 2000 | 0 |
| 4000 | 299 |
| 8000 | 801 |
| 16000 | 2397 |
| 32000 | 7679 |
| 64000 | 25731 |
| 128000 | 92453 |
| 256000 | 311468 |
|  |  |

For 4000 vertices:

|  |  |
| --- | --- |
| Edges | Time(micro sec) |
| 4000 | 0 |
| 8000 | 601 |
| 16000 | 1795 |
| 32000 | 5286 |
| 64000 | 16954 |
| 128000 | 54753 |
| 256000 | 192385 |
| 512000 | 720571 |
| 1024000 | 2644726 |

For 8000 vertices:

|  |  |
| --- | --- |
| Edges | Time(micro sec) |
| 8000 | 0 |
| 16000 | 1994 |
| 32000 | 3690 |
| 64000 | 10571 |
| 128000 | 34009 |
| 256000 | 132342 |
| 512000 | 497071 |
| 1024000 | 1694165 |
| 2048000 | 5898089 |
| 4096000 | 23054602 |

For 16000 vertices:

|  |  |
| --- | --- |
| Edges | Time(micro sec) |
| 16000 | 98 |
| 32000 | 2892 |
| 64000 | 7878 |
| 128000 | 22739 |
| 256000 | 74320 |
| 512000 | 376577 |
| 1024000 | 1080094 |
| 2048000 | 4247429 |
| 4096000 | 14804382 |

Report on runtimes of BFS on Adjacency Matrix implementation of Graph:

For 1000 vertices:

|  |  |
| --- | --- |
| Edges Count | Time(micro sec) |
| 1000 | 5282 |
| 2000 | 7779 |
| 4000 | 7882 |
| 8000 | 7879 |
| 16000 | 8481 |
| 32000 | 10872 |
| 64000 | 14860 |

For 2000 vertices:

|  |  |
| --- | --- |
| Edges Count | Time(micro sec) |
| 2000 | 67818 |
| 4000 | 119480 |
| 8000 | 123071 |
| 16000 | 111302 |
| 32000 | 114194 |
| 64000 | 105515 |
| 128000 | 94650 |
| 256000 | 84271 |

For 4000 vertices:

|  |  |
| --- | --- |
| Edges | Time(micro sec) |
| 4000 | 232179 |
| 8000 | 555064 |
| 16000 | 569248 |
| 32000 | 518224 |
| 64000 | 505938 |
| 128000 | 507624 |
| 256000 | 506737 |
| 512000 | 494585 |
| 1024000 | 467177 |

For 8000 vertices:

|  |  |
| --- | --- |
| Edges | Time(micro sec) |
| 8000 | 1476577 |
| 16000 | 1875841 |
| 32000 | 2061539 |
| 64000 | 2208378 |
| 128000 | 1973197 |
| 256000 | 2029607 |
| 512000 | 1809890 |
| 1024000 | 1595784 |
| 2048000 | 1618761 |
| 4096000 | 1706999 |

sFor 16000 vertices

|  |  |
| --- | --- |
| Edges | Time(micro sec) |
| 16000 | 5018001 |
| 32000 | 9076229 |
| 64000 | 8530962 |
| 128000 | 8148929 |
| 256000 | 8009040 |
| 512000 | 8067524 |
| 1024000 | 8345708 |
| 2048000 | 8525458 |
| 4096000 | 7740738 |
| 8192000 | 6542908 |
| 16384000 | 6439610 |

**Question Answer:**

1. What is the impact on runtime if we keep |V| unchanged and double |E| for adjacency list? Why is it so?

**Answer**: From the statistics Table of adjacency list,we can see that if we keep |V| constant and double |E| the runtime increases.We know that the runtime of BFS of a graph with |V| vertices and |E| edges is O(V+E).Now if |V| is less than |E| ,then |V|+|E|<=|E|+|E|=2|E|.As we ignore constant factors in asymptotic notation ,we see that when |E| >|V| .O(V+E) really means O(E). so if we increase |E| keeping |V| unchanged the runtime will increase proportionally.

2. What is the impact on runtime if we keep |E| unchanged and double |V| for adjacency list? Why is it so?

**Answer**: From the statistics table of adjacency list ,we can see that if we keep |E| constant and double |V| the runtime increases. We know that the runtime of BFS of a graph with |V| vertices and |E| edges is O(V+E) .Now if |E| is less than |V| , then |V|+|V|=2|V|.As we ignore constant factors in asymptotic notation.we see that when |V|>|E| .O(V+E) really means O(V). so if we increase |V| keeping |E| unchanged the runtime will increase proportionally.

1. What is the impact on runtime if we keep |V| unchanged and double |E| for adjacency matrix? Why is it so?

**Answer**: From the statistics Table of adjacency matrix ,we can see that if we keep |V| constant and double |E| the runtime remains almost the same . This is because for adjacency matrix , the time complexity of bfs is always O(n^2),where n is the number of vertices in a graph . So ,the runtime is not really affected if we double the number of edges.

1. What is the impact on runtime if we keep |E| unchanged and double |V| for adjacency matrix? Why is it so?

**Answer:** From the Statistics Table of adjacency matrix , we can see that if we keep |E| constant and double |V| the runtime increases. This is because for adjacency matrix, the time complexity of bfs as always O(n^2) . where n is the number of vertices in a graph .So the runtime will increases if we double the number of vvertices.

1. For the same |E| and |V|, why are the runtimes for adjacency list and adjacency matrix representation different? Which one is higher and why?

**Answer**:For the same |E| and |V| ,runtime of adjacency matrix is higher than that of adjacency list.But when |E| is very high,the runtime of BFS for both of them are not that much different .That means if the graph is dense,runtime of BFS for adjancency matrix and list is almost the same .This is because the time complexity of adjacency matrix is always O(n^2) and in this case time complexity of adjacency list is O(n^2+n) =O(n^2) ,where m=|V| and n=|E| .But if the graph is sparse that means |E| is low the runtime of bfs for adjacency list is not lower than that of adjacency matrix .This is because in this case the time complexity of adjacency list of O(m+n) =O(m),on the other hand for adjacency matrix it is always O(n^2) .Here m=|V| and n=|E|.