Student ID: 122460582

Version 2: evaluating Prim's algorithm with different priority queue implementations on random graphs.

Variation 1:

I have ran multiple tests on graph creation and eventually picked one that created a graph in the fastest time due to the reduce amount of randomizing, I have including both graph generation methods within my code file. For the first test, I used the imported heapq from python's built-in library and also used lambda within the method, then my outcomes were the following:

```
Vertices:  1000
Generating high density graph with 90% of the maximum number of edges
Graph generated(25.758732700007386 seconds)
Prim APQ Heap on high density graph of 90% :(26.66917249999824 seconds)
Prim APQ Unsorted List on high density graph of 90% : (0.23314559999562334 seconds)
```

```
Vertices:  1000
Generating low density graph with 25% of the maximum number of edges
Graph generated(1.9038641000079224 seconds)
Prim APQ Heap on low density graph of 25% :(2.0998772000020836 seconds)
Prim APQ Unsorted List on low density graph of 25% : (0.08579989999998361 seconds)
```

This demonstrates that Prim APQ Unsorted List was better in a sparse and dense graph of vertices size 1000. When I checked back the runtime of this code, I found that the lambda was causing the heap to have an extremely higher time compared to list.

Variation 2:

This is using code that was implemented by someone else without using any in-built libraries:

```
Unsorted List Sparse, Size=4000, Time=0.18988632700003108
Heap Sparse, Size=4000, Time=0.027506973999934417
```

```
Unsorted List Dense, Size=4000, Time=1.8498529409999447
Heap Dense, Size=4000, Time=1.876504381999921
```

From this we can see that with a vertices size of 4000, heap is better than unsorted list for a sparse graph. It also shows that heap is better for a dense graph but the time difference closing up between the both for a dense graph. We can conclude that in a scenario where if

both methods are implemented without using any built in functions or lambda, unsorted list will be better than heap for a dense graph if the vertices size are around 10000 or more.

Variation 3 (Final Variation) :

This is my final variation of the code, to complete the code I received helped by using online codes and ChatGPT to fix my codes since judging by the previous codes, some of my codes were inefficient causing such a high runtime (can be seen in variation 1). Then in variation 2, I took code from online resources and ChatGPT. After attempting variation 2 I realized my variation 1 implementation was wrong, hence I implemented variation 3 after testing out the outputs for variation 2. I slightly adjusted the graph function to reduce the time spent generating the graph to test on a larger graph vertices size, since I saw a pattern after testing variation 2. I saw that the times of unsorted list for a dense graph on a larger amount of vertices came extremely close to a heap on a dense graph. This attached screenshot was from variation 2, as seen for low amount of vertices, heap was a lot better than unsorted list but when I increased the vertices to 4000, it became really close hence to allow me to test on a larger vertices size, I had to slightly readjust my graph generation function to reduce the hours spent generating a graph.

```
Unsorted List Dense, Size=10, Time=2.6008000000664094e-05
Heap Dense, Size=10, Time=3.670100001727406e-05
```

The screenshots below are my results for my final variation, where I combined my original code with ChatGPT codes to implement the final variation, where both of the methods did not use any imports, which in return gave me a more accurate result I believe.

```
--------------------------------------------------------------------------------
Prim APQ Heap on dense graph of 100 vertices: 0.0031457999866688624 seconds
Prim APQ List on dense graph of 100 vertices: 0.0025163999962387607 seconds
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
Prim APQ Heap on dense graph of 200 vertices: 0.015177799999946728 seconds
Prim APQ List on dense graph of 200 vertices: 0.011464999988675117 seconds
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
Prim APQ Heap on dense graph of 500 vertices: 0.07983509999758098 seconds
Prim APQ List on dense graph of 500 vertices: 0.07921169999463018 seconds
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
Prim APQ Heap on dense graph of 1000 vertices: 0.31910840001364704 seconds
Prim APQ List on dense graph of 1000 vertices: 0.31291190000774805 seconds
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
Prim APQ Heap on dense graph of 2000 vertices: 1.3101654000056442 seconds
Prim APQ List on dense graph of 2000 vertices: 1.3262006000004476 seconds
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
Prim APQ Heap on dense graph of 5000 vertices: 8.43893959998968 seconds
Prim APQ List on dense graph of 5000 vertices: 8.410247599997092 seconds
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
Prim APQ Heap on dense graph of 10000 vertices: 35.175035200009006 seconds
Prim APQ List on dense graph of 10000 vertices: 34.12186329999531 seconds
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
Prim APQ Heap on dense graph of 12000 vertices: 67.57355919999827 seconds
Prim APQ List on dense graph of 12000 vertices: 49.17905799999426 seconds
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
Prim APQ Heap on dense graph of 14000 vertices: 207.2398181000026 seconds
Prim APQ List on dense graph of 14000 vertices: 66.04595329999574 seconds
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
Prim APQ Heap on dense graph of 16000 vertices: 551.6609912999993 seconds
Prim APQ List on dense graph of 16000 vertices: 188.27742889999354 seconds
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
Prim APQ Heap on dense graph of 20000 vertices: 3840.107777199999 seconds
Prim APQ List on dense graph of 20000 vertices: 890.5449621999869 seconds
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
```

```
--------------------------------------------------------------------------------
Prim APQ Heap on sparse graph of 100 vertices: 0.0010871000122278929 seconds
Prim APQ List on sparse graph of 100 vertices: 0.0004547999971576035 seconds
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
Prim APQ Heap on sparse graph of 200 vertices: 0.0018133000121451914 seconds
Prim APQ List on sparse graph of 200 vertices: 0.0013635999930556864 seconds
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
Prim APQ Heap on sparse graph of 500 vertices: 0.006533499981742352 seconds
Prim APQ List on sparse graph of 500 vertices: 0.0071712000062689185 seconds
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
Prim APQ Heap on sparse graph of 1000 vertices: 0.010818400012794882 seconds
Prim APQ List on sparse graph of 1000 vertices: 0.02417370001785457 seconds
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
Prim APQ Heap on sparse graph of 2000 vertices: 0.030384099984075874 seconds
Prim APQ List on sparse graph of 2000 vertices: 0.09318869997514412 seconds
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
Prim APQ Heap on sparse graph of 5000 vertices: 0.063650800002506 seconds
Prim APQ List on sparse graph of 5000 vertices: 0.6625648000044748 seconds
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
Prim APQ Heap on sparse graph of 10000 vertices: 0.17504870000993833 seconds
Prim APQ List on sparse graph of 10000 vertices: 2.866672000003746 seconds
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
Prim APQ Heap on sparse graph of 12000 vertices: 0.1857179999933578 seconds
Prim APQ List on sparse graph of 12000 vertices: 4.189661999989767 seconds
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
Prim APQ Heap on sparse graph of 14000 vertices: 0.24000180000439286 seconds
Prim APQ List on sparse graph of 14000 vertices: 5.714459599985275 seconds
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
Prim APQ Heap on sparse graph of 16000 vertices: 0.31045689998427406 seconds
Prim APQ List on sparse graph of 16000 vertices: 7.848947399994358 seconds
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
Prim APQ Heap on sparse graph of 20000 vertices: 0.3479114000219852 seconds
Prim APQ List on sparse graph of 20000 vertices: 12.652747899992391 seconds
--------------------------------------------------------------------------------
```

Table presentation of my final variation:

Vertices(n)	APQ List (Dense)	APQ Heap (Dense)
100	0.0025163999962387607 seconds	0.0031457999866688624 seconds
200	0.011464999988675117 seconds	0.015177799999946728 seconds
500	0.07921169999463018 seconds	0.07983509999758098 seconds
1000	0.31291190000774805 seconds	0.31910840001364704 seconds
2000	1.3262006000004476 seconds	1.3101654000056442 seconds
5000	8.410247599997092 seconds	8.43893959998968 seconds
10000	34.12186329999531 seconds	35.175035200009006 seconds
12000	49.17905799999426 seconds	67.57355919999827 seconds
14000	66.04595329999574 seconds	207.2398181000026 seconds
16000	188.27742889999354 seconds	551.6609912999993 seconds
20000	890.5449621999869 seconds	3840.107777199999 seconds

Vertices(n)	APQ List (Sparse)	APQ Heap (Sparse)
100	0.00045479999971576035 seconds	0.0010871000122278929 seconds
200	0.0013635999930556864 seconds	0.0018133000121451914 seconds
500	0.007171200006268918 5 seconds	0.006533499981742352 seconds
1000	0.02417370001785457 seconds	0.010818400012794882 seconds
2000	0.09318869997514412 seconds	0.030384099984075874 seconds
5000	0.6625648000044748 seconds	0.063650800002506 seconds
10000	2.866672000003746 seconds	0.17504870000993833 seconds
12000	4.189661999989767 seconds	0.1857179999933578 seconds
14000	5.714459599985275 seconds	0.24000180000439286 seconds
16000	7.848947399994358 seconds	0.31045689998427406 seconds
20000	12.652747899992391 seconds	0.3479114000219852 seconds

Using the data from my testing, we can see after size 10000 for vertices for a dense graph, APQ List (unsorted list) outperforms APQ Heap by a lot. Especially when I doubled the vertices to 20000 for the dense graph, the time difference is almost 3000 seconds. From this we can conclude that APQ Heap is better than APQ List (unsorted list) on a dense graph if vertices is a small size, around 2000 or less, but APQ List (unsorted list) is a lot bigger than APQ Heap on a dense graph with a large vertices size, size of around 10000 or more. APQ Heap is significantly better than APQ List (unsorted) if vertices is around 14000 or more. We can see massive time gap between the two.

Conclusion for dense graphs:
APQ List (unsorted list) is better than APQ Heap overall, and it is significantly better than APQ Heap at a large vertices size at around 14000.

For a sparse graph, APQ Heap was a lot better at a larger vertices size. At a size of 20000, APQ Heap was still under 1 second in comparison to APQ List (unsorted list) which was at 12.652747899992391 seconds. At a size of 100 and 200, APQ List (unsorted list) was slightly better than APQ Heap but after size of 500, APQ Heap was a lot better than APQ List (unsorted list).

Conclusion: APQ Heap is better than APQ List (unsorted list) for a sparse graph at a vertices size of 500 or more and it is significantly better than APQ List (unsorted list) at a vertices size of 10000 or more, as it was under 1s for 20000 in comparison to APQ List (unsorted) that was over 12s. Even at 10000, it was almost just slightly under 3s for APQ List (unsorted) but it was just 0.17504870000993833 seconds for APQ Heap. Based on my testing and implementation, APQ List (unsorted list) is better than APQ Heap for a dense graph (90%+) and APQ Heap is better than APQ List (unsorted list) for a sparse graph (with minimum required edges).

To make sure my results were accurate, I ran the same code on my home computer to make sure the times returned for certain method is higher than another.

Results for a sparse graph:

```
--------------------------------------------------------------------------------
Prim APQ Heap on sparse graph of 100 vertices: 0.08855859999312088 seconds
Prim APQ List on sparse graph of 100 vertices: 0.0012939999578520656 seconds
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
Prim APQ Heap on sparse graph of 200 vertices: 0.006666900007985532 seconds
Prim APQ List on sparse graph of 200 vertices: 0.0022221000399440527 seconds
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
Prim APQ Heap on sparse graph of 500 vertices: 0.0055964000057429075 seconds
Prim APQ List on sparse graph of 500 vertices: 0.005046000005677342 seconds
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
Prim APQ Heap on sparse graph of 1000 vertices: 0.009425199998077005 seconds
Prim APQ List on sparse graph of 1000 vertices: 0.013717499969061464 seconds
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
Prim APQ Heap on sparse graph of 2000 vertices: 0.017230199999175966 seconds
Prim APQ List on sparse graph of 2000 vertices: 0.04963060002774 seconds
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
Prim APQ Heap on sparse graph of 5000 vertices: 0.03964589996030554 seconds
Prim APQ List on sparse graph of 5000 vertices: 0.2887544999830425 seconds
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
Prim APQ Heap on sparse graph of 10000 vertices: 0.08585590001894161 seconds
Prim APQ List on sparse graph of 10000 vertices: 1.220870599965565 seconds
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
Prim APQ Heap on sparse graph of 12000 vertices: 0.1065072999917902 seconds
Prim APQ List on sparse graph of 12000 vertices: 1.7685328000225127 seconds
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
Prim APQ Heap on sparse graph of 14000 vertices: 0.11735040001804009 seconds
Prim APQ List on sparse graph of 14000 vertices: 2.3938945999834687 seconds
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
Prim APQ Heap on sparse graph of 16000 vertices: 0.1401960999937728 seconds
Prim APQ List on sparse graph of 16000 vertices: 3.2114355000085197 seconds
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
Prim APQ Heap on sparse graph of 20000 vertices: 0.1798001999850385 seconds
Prim APQ List on sparse graph of 20000 vertices: 5.039356300025247 seconds
--------------------------------------------------------------------------------
```

Results for a dense graph:

```
--------------------------------------------------------------------------------
Prim APQ Heap on dense graph of 100 vertices: 0.0011367000406607985 seconds
Prim APQ List on dense graph of 100 vertices: 0.001020299969241023 seconds
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
Prim APQ Heap on dense graph of 200 vertices: 0.003757600032258779 seconds
Prim APQ List on dense graph of 200 vertices: 0.0038968000444583595 seconds
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
Prim APQ Heap on dense graph of 500 vertices: 0.027067799994256347 seconds
Prim APQ List on dense graph of 500 vertices: 0.023334499972406775 seconds
--------------------------------------------------------------------------------
```

```
--------------------------------------------------------------------------------
Prim APQ Heap on dense graph of 1000 vertices: 0.10768549999920651 seconds
Prim APQ List on dense graph of 1000 vertices: 0.10374220000812784 seconds
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
Prim APQ Heap on dense graph of 2000 vertices: 0.3940853000385687 seconds
Prim APQ List on dense graph of 2000 vertices: 0.4084234000183642 seconds
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
Prim APQ Heap on dense graph of 5000 vertices: 2.6516697999904864 seconds
Prim APQ List on dense graph of 5000 vertices: 2.4804422999732196 seconds
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
Prim APQ Heap on dense graph of 10000 vertices: 11.49154630000703 seconds
Prim APQ List on dense graph of 10000 vertices: 10.389979099971242 seconds
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
Prim APQ Heap on dense graph of 12000 vertices: 21.89592730003642 seconds
Prim APQ List on dense graph of 12000 vertices: 14.707366899994668 seconds
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
Prim APQ Heap on dense graph of 14000 vertices: 42.67489740002202 seconds
Prim APQ List on dense graph of 14000 vertices: 24.572434499976225 seconds
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
Prim APQ Heap on dense graph of 16000 vertices: 89.94033680000575 seconds
Prim APQ List on dense graph of 16000 vertices: 38.42312349996064 seconds
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
Prim APQ Heap on dense graph of 20000 vertices: 309.06301009998424 seconds
Prim APQ List on dense graph of 20000 vertices: 103.86995799996657 seconds
--------------------------------------------------------------------------------
```

Looking at these outputs on another computer, my conclusion from earlier has been proven, that based on my testing and implementation, APQ List (unsorted list) is better than APQ Heap for a dense graph (90%+) and APQ Heap is better than APQ List (unsorted list) for a sparse graph (with minimum required edges). Looking from these outputs, this conclusion proves to be true.