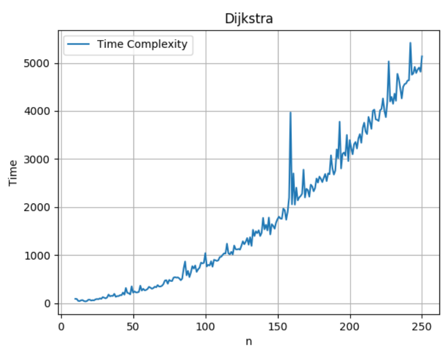
HW5

Miguel A. Montoya Z. (A01226045)

November 17, 2017

1. Implement Dijkstra Algorithm

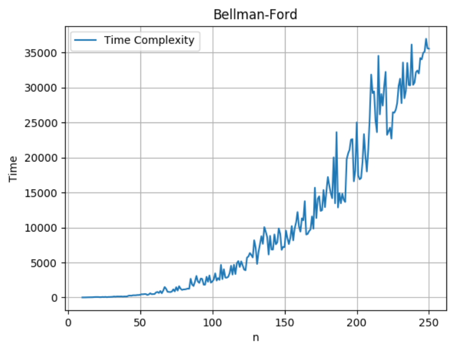
   Dijkstra algorithm takes $O(E + V\ lg(V))$.
   The following graph represents the time taken to execute this algorithm in a randomly generated complete graph with E edges, only if the given alpha is bigger than generated alpha. The implementation resembles the theoretical time complexity of the Dijkstra algorithm after the increase of size.

   The code used is annexed in **Dijkstra.cpp**

2. Implement Bellman-Ford Algorithm

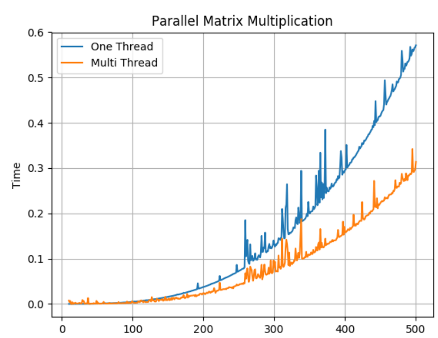
Bellman-Ford algorithm takes $O(V\ lg(V))$.
The following graph represents the time taken to execute this algorithm in a randomly generated complete graph with V vertices, as before, only if the given alpha is bigger than generated alpha. The implementation resembles the theoretical time complexity of the Bellman-Ford algorithm after the increase of size.



The code used is annexed in **Bellman.cpp**

3. Implement the Parallel Matrix Multiplication using OpenMP

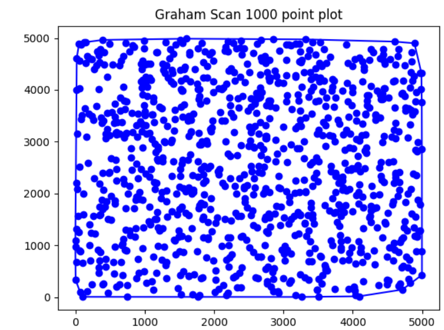
   The following graph represents the time taken to execute this algorithm in a randomly generated matrix NxN. Two lines, one for a single thread multiplication, and the other one for a multi thread multiplication of the same matrix. The implementation behaves as expected, and a multi threaded operations is executed faster.



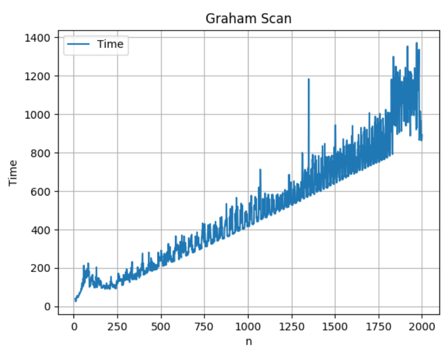
   The code used is annexed in **ParallelMM.cpp**

4. Implement the Graham Scan Algorithm

After the execution of the Graham Scan with 1000 points, the results were successful.



Graham scan algorithm takes $O(V\ lg(V))$.
The following graph represents the time taken to execute this algorithm with randomly generated points. The implementation behaves as expected.



The code used is annexed in **ParallelMM.cpp**

5. STINGY SAT is the following problem:

   Given a set of clauses (each a disjunction of literals) and an integer k, find athat satisfying assignment in which at most k variables are true, if such an assignment exists
   STINGY SAT is NP-complete.

   1.- Prove that Stingy SAT is NP:
   Given a solution to STINGY SAT, its validity can be checked by evaluating. The check for less than k variables are true by checking each and counting.
   2.- SAT can be reduced to STINGY SAT.
   For each accepted formula f of SAT with at most n true variables, a configuration for STINGY SAT (f, n) is accepted.
   Any accepted configuration (f, n) of STINGY SAT would provide a solution for SAT with the formula f.

6. The set-partition problem is the following:

   The set-partition problem takes as input a set S of numbers. The questionthat is whether S can be partitioned into two sets A and $\overline{A} = S - A$ such that $\sum_{x \in A} x = \sum_{x \in \overline{A}} x$
   the set-partition problem is NP-complete.

   1.- Prove that set-partition is NP:
   Given two sets, the problem can be verified by comparing the two total sums.
   2.- SUBSET-SUM can be reduced to SET-PARTITION
   Being sum the total sum of elements in X and $target$ the target total of SUBSET-SUM, we can declare $X' = X \cup \{sum - 2target\}$ making the total of $X' = 2sum - 2target$, and feed that to SET-PARTITION.
   If the configuration $(X, target)$ is accepted in SUBSET-SUM, SUBSET-PARTITION must accept X'. Because the remaining numbers in X would be $sum - target$, this would imply that X' can be partitioned into to sets that each sums to $sum - target$.
   If a partition of X' can be made, each partition would have a total sum of $sum - target$. One of these sets has the number $sum - 2target$, removing it would provide an accepted configuration for SUBSET-SUM.

7. Show that the following is NP-complete.

   INPUT: Two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$; a budget b.
   OUTPUT: Two sets of nodes $V''_1 \subseteq V_1$ and $V'_2 \subseteq V_2$ whose deletion leaves at least b nodes in each graph, and makes the two graphs identical.

   1.- Prove that two-graphs problem is NP:
   Given two sets to remove, the configuration can be proven if the remaining nodes in a graph can each be mapped to the other one.