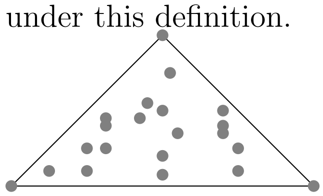
COMP3011 Homework 4

ZHANG Caiqi 18085481d

December 7, 2020

1

Let us firstly define how to arrange the points. For the n points, we choose three points to form a triangle and let the rest of the points inside this triangle. The following is an example under this definition.



The Graham's scan takes $O(nlgn)$ time to get the convex hull. The Jarvis's match takes $O(3n)$ time to calculate the convex hull. Therefore, Jarvis's march is asymptotically faster than Graham's scan.
The reason why Jarvis's march is faster in this particular case is that the number of points in convex hull in this case is relatively small, which is 3. For Jarvis's march, the time complexity is $O(nh)$, when h is small, the performance will be better.

2

The basic idea of my algorithm is that firstly, randomly choose a point as P_0, then calculate the slope of every other points with P_0 and stored the result to a hashmap. If we find any two slopes are the same, then there are three points that are collinear. After we choose all the points as P_0 and still cannot find three collinear points, then the algorithm output "NO".
The Pseudo-code is as follows.
About the time complexity, in the worst situation, we will choose all the points in Q as P_0. And in each iteration, calculating and checking the slope one one by one takes $O(n)$ time. Operations of Hashmap takes $O(1)$ time. Therefore, the overall time complexity is $O(n^2)$.

Algorithm 1: Find Collinear

Input: A sequence Q of points in the plane.
Output: "YES" if Q contains three points that are collinear; "NO" otherwise.

Initialize HashMap;
for P_0 *in* Q **do**
    for P *in* Q *and* $P \neq P_0$ **do**
        HashMap.put(Slope(P, P_0))
        if *collision* **then**
            return "YES"
        end
    end
end
return NO

3

We can try to give the proof based on David Maier's [1] proof in 1978 on showing ULCS is NP-complete by node cover problem. To be consistant of the paper, here we will use the notation mentioned in the paper.

- Firstly, ULCS is a NP problem. For any candidate answer, we can easily check whether all the sequences contain this common subsequence and whether this subsequence has correct length. This takes polynomial time.

- Secondly, to prove that each problem in NP can be reduced to ULCS, use the following polynomial-time reduction from Independent-Set:

  When we have a graph $G(N, E)$, all the edges can be described in (x_0, y_0); (x_2, y_2); ...; (x_r, y_r). For the nodes, we assign a random order to it, say $\{v_1, v_2, ..., v_t\}$. Then, we will construct $(r+1)$ sequences of length at most 2(t - 1) as shown in the following figure. The first sequence is the template T, which is exactly the all the nodes in previous assigned order $\{v_1, v_2, ..., v_t\}$. For the other r sequences, for each edge $e_i = (x_i, y_i)$ in E we construct a sequence S_i. Assume without loss of generality that $x_i = v_j$, $y_i = v_m$, and $j < m$. Then S_i, is $v_1, v_2, ..., v_{j-1}, v_{j+1}, ..., v_{m-1}, v_{m+1}, ..., v_t$.

  Now let's prove the most significant part of the proof, which is *The graph G has a independent set of size k only if the $R = \{T, S_1, S_2, ..., S_t\}$ has a common subsequence of size k.* Let $U = \{u_1, u_2, ..., u_k\}$, For any edge $\{x_i, y_i\}$ containing one node in U, let $v_j = x_i$ and $v_m = y_i$. If v_j not in U, U is the subsequence of first-half of S_i. If v_j not in U, U is the subsequence of the second-half of S_i. Therefore, in any case, U is the subsequence of R.

- Therefore, ULCS is NP-complete.

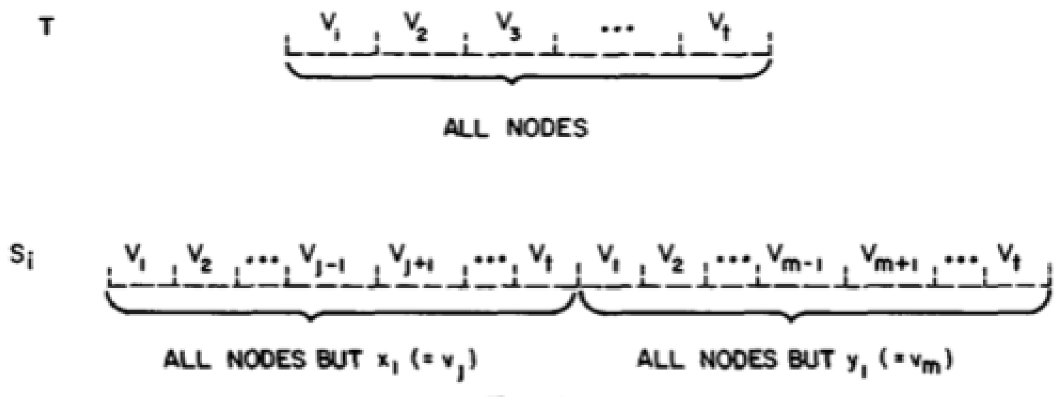


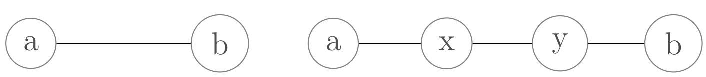
Figure 1: T and S_i

4

The independent set problem under the triangle free restriction is also NP-complete problem. We will prove this mainly based on Svatopluk Poljak's [2] work on 1974.

- Firstly, independent set problem on triangle free graph is a NP problem. For any candidate answer, we can easily check whether it satisfies the requirement of independent set. This takes polynomial time.
- Secondly, to prove that each problem in NP can be reduced to independent set problem on triangle free graph, use the following polynomial-time reduction from Independent-Set:

  Suppose that G is a graph with n vertices and m edges. Let $\alpha(G)$ denote the maximum independent set size of G. Then, replace each edge of G by a path consisting of three edges. The following graph give an example of edge {a, b}.

  We call the new graph F. Then F has $n + 2m$ vertices and $\alpha(F) = \alpha(G) + m$. This is because of edge {a, b}, let any endpoint in the independent set, say a, then after reconstruction, there will be a new node in the independent set, in this case, which is y. As F is triangle free, this is the reduction of G to a triangle free graph F, which can be done in polynomial time.
- Therefore, the independent set problem under the triangle free restriction is also NP-complete problem.

5

No. We prove by contradiction. If there is a decision problem, such that it belongs to P and also NPC. Because it is NPC, all the other NP problem can be reduced to this decision problem. Also, it is P problem, which means it can be solved in polynomial time. Then, it shows that $P = NP$. Contradiction. Therefore, there is no such decision problem when $P \neq NP$.

6

Firstly, we try to show how to reduce this problem into *Minimum Vertex Cover Problem.* The method is that we firstly use the Approx-Vertex-Cover algorithm to find the vertex cover set G, then assign all the edges a direction to the endpoints not in G. As G covers all the edges, after assignment the graph is a directed graph such that the number of vertices with at least one outgoing edge is as small as possible.
According to the *Theorem 35.1* in lecture 12, Approx-Vertex-Cover algorithm is a polynomial-time 2-approximation algorithm. The approximation ratio is:

$$\frac{|C|}{|C^*|} \leq \frac{2|C^*|}{|C^*|} = 2$$

References

[1] D. Maier, "The complexity of some problems on subsequences and supersequences," *Journal of the ACM (JACM)*, vol. 25, no. 2, pp. 322–336, 1978.

[2] S. Poljak, "A note on stable sets and colorings of graphs," *Commentationes Mathematicae Universitatis Carolinae*, vol. 15, no. 2, pp. 307–309, 1974.