- Always try to give algorithm with best possible running time. The points that you obtain will depend on the running time of your algorithm. For example, a student who gives an $O(n)$ algorithm will receive more points than a student who gives an $O(n^2)$ algorithm.

- **Use of unfair means will be severely penalized.**

There are 1 questions for a total of 50 points.

---

1. This is question 26-6 from the CLRS book. Here is the description of the problem in case you do not have a copy of the book.

   In this problem, we describe a faster algorithm, due to Hopcroft and Karp, for finding a maximum matching in a bipartite graph. The algorithm runs in $O(\sqrt{n} \cdot m)$ time. Given an undirected, bipartite graph $G = (V, E)$, where $V = L \cup R$ and all edges have exactly one endpoint in L, let M be a matching in G. We say that a simple path P in G is an *augmenting path* with respect to M if it starts at an unmatched vertex in L, ends at an unmatched vertex in R, and its edges belong alternately to M and $E - M$. (This definition of an augmenting path is related to, but different from, an augmenting path in a flow network.) In this problem, we treat a path as a sequence of edges, rather than as a sequence of vertices. A shortest augmenting path with respect to a matching M is an augmenting path with a minimum number of edges.

   Given two sets A and B, the *symmetric difference* $A \oplus B$ is defined as $(A - B) \cup (B - A)$, that is, the elements that are in exactly one of the two sets.

(5) (a) Show that if M is a matching and P is an augmenting path with respect to M, then the symmetric difference $M \oplus P$ is a matching and $|M \oplus P| = |M| + 1$. Show that if $P_1, ..., P_k$ are vertex-disjoint augmenting paths with respect to M, then the symmetric difference $M \oplus (P_1 \cup ... \cup P_k)$ is a matching with cardinality $|M| + k$.

   The general structure of our algorithm is the following:

   HOPCROFT-KARP(G)

   1. $M = \{\}$
   2. **repeat**
   3. Let $\mathcal{P} = \{P_1, ..., P_k\}$ be a maximal set of vertex-disjoint shortest
   . augmenting paths with respect to M.
   4. $M = M \oplus (P_1 \cup ... \cup P_k)$
   5. **until** $|\mathcal{P}| == 0$
   6. **return** M

   The remainder of this problem asks you to analyse the number of iterations in the algorithm (that is, the number of iterations in the **repeat** loop) and to describe an implementation of line 3.

(5) (b) Given two matchings M and M^* in G, show that every vertex in the graph $G' = (V, M \oplus M^*)$ has degree at most 2. Conclude that G' is a disjoint union of simple paths or cycles. Argue that edges in each simple path or cycle belongs alternately to M or M^*. Prove that if $|M| \leq |M^*|$, then $M \oplus M^*$ contains at least $|M^*| - |M|$ vertex-disjoint augmenting paths with respect to M.

Let l be the length of a shortest augmenting path with respect to a matching M, and let $P_1, ..., P_k$ be a maximal set of vertex-disjoint augmenting paths of length l with respect to M. Let $M' = M \oplus (P_1 \cup ... \cup P_k)$, and suppose that P is a shortest augmenting path with respect to M'.

(5) (c) Show that if P is vertex-disjoint from $P_1, ..., P_k$, then P has more than l edges.

(10) (d) Now suppose that P is not vertex-disjoint from $P_1, ..., P_k$. Let A be the set of edges $(M \oplus M') \oplus P$. Show that $A = (P_1 \cup ... \cup P_k) \oplus P$ and that $|A| \geq (k+1)l$. Conclude that P has more than l edges.

(5) (e) Prove that if a shortest augmenting path with respect to M has l edges, the size of the maximum matching is at most $|M| + n/(l+1)$.

(10) (f) Show that the number of **repeat** loop iterations in the algorithm is at most $2\sqrt{n}$. (*Hint*: By how much can M grow after iteration number $\sqrt{n}$?)

(10) (g) Give an algorithm that runs in $O(m)$ time to find a maximal set of vertex-disjoint shortest augmenting paths $P_1, ..., P_k$ for a given matching M. Conclude that the total running time of HOPCROFT-KARP is $O(\sqrt{n}m)$.