Homework Assignment 3

Matthieu Boyer

Contents

1 Exercise 1 **2**

2 Exercise 2 **3**
2.1 Question 1 3
2.2 Question 2 3
2.3 Question 3 3
2.4 Question 4 3

3 Exercise 3 **4**
3.1 Question 1 4
3.2 Question 2 4

1 Exercise 1

To show that this algorithm works, it suffices to show that the union operation is correct : We will denote by X_i and X_j the sets in which i and j are.
First, it is clear that from the line `if id[i] == id[k]`, if $k \notin X_i$, `id[k]` is not modified and thus X_k is not modified.
Since this is also true, we only need to show that all elements in X_i are put in X_j meaning that $\forall k \in X_i,$ `id[k]` $=$ `id[j]` after the operation. Let us denote r_i the value of `id[i]` before the operation and let $r_j =$ `id[j]`.
If $X_i = X_j$, we do have this result, however, in all generality if $i \neq \max X_i$, this will not follow. Indeed then, if $i < l \in X_i$, when arriving to the step i in the `for` loop, since `id[i]` $=$ `id[i]`, we set `id[i]` to `id[j]`. Then, when arriving to the step l, since at this point `id[l]` $= r_i \neq r_j =$ `id[i]`, l will not be added to X_j.

We could fix this algorithm by not checking the value of `id[i]` at each step in the loop, but by setting its value in a variable before.

2 Exercise 2

2.1 Question 1

- First, we will show that the minimum spanning tree (MST) of G is unique. Suppose that there are two different MSTs A and B. Let $u \in A \Delta B$ be of minimum weight. Without loss of generality, we suppose $u \in A$. Since B is a tree, $u \cup B$ has a cycle C containing u. Since A is a tree and C is a cycle, there is an edge v in C that is not in A. As u is of minimum weight, and since the weight are all distinct : $w(v) > w(u)$. Since C is a cycle, removing v from B to replace it with u yields a new weighted spanning tree with weight lower than B, which is a contradiction.

- Then, to see that there can be two second best spanning tree, take the weighted graph over $[\![1, 4]\!]$, represented below by adjacence matrix :

$$\begin{pmatrix} \infty & 1 & 3 & 4 \\ 1 & \infty & 2 & 5 \\ 3 & 2 & \infty & \infty \\ 4 & 5 & \infty & \infty \end{pmatrix}$$

2.2 Question 2

Since any spanning tree has $|V|-1$ edges by the properties of trees, any second-best MST (SBMST) must have at least one edge that is not in the MST (it is from question 1 unique). If a SBMST has exactly one edge $xy \notin E(T)$, then, it has the same set of edges as the MST with xy replacing some edge uv of the MST. In this case, our SBMST $T^{'} = T - uv + xy$. We thus have showed that all SBMST can be found by swapping one edge from the MST. Since the set of all spanning trees has at least two elements as $|E| \geq |V| > |V| - 1$ with this last number being the number of edges in a spanning tree. Thus, there is a pair of edge $xy \notin E(T)$, $uv \in E(T)$ such that $T - uv + xy$ is a SBMST of G.

2.3 Question 3

To fill in $\max[u, v]$ for all u, v, we only need to do a search in T from each node u. We do $|V|$ such searches, and those are done in linear time in $\mathcal{O}(|V| + |E(T)|) = \mathcal{O}(|V| + |V| - 1) = \mathcal{O}(|V|)$, thus we get a total time complexity in $\mathcal{O}(|V|^2)$.

2.4 Question 4

We first can compute a minimum spanning tree T in $\mathcal{O}(|V|^2)$. Then, from 3), we compute the max table in time $\mathcal{O}(V^2)$. We then try to minimize for x, y an edge not in T the quantity : $w\left(u, v\right) - w(x, y)$ where $\max[x, y] = u, v$. Indeed, if we create $T^{'} = T + xy - uv$, we minimize $w(T^{'})$ and make $T^{'}$ a SMBST (since it cannot be a MST) if and only if the previous quantity is minimized. We then return $T^{'}$. Finally, we get an $\mathcal{O}(\left|V^2\right|)$ algorithm to compute a second-best minimum spanning tree.

3 Exercise 3

3.1 Question 1

We have $|G| = 2n$. Since M is a matching, for $(x, y) \in M$, we cannot have both x and y in a stable set of G. Then, since M covers $2k$ of these vertices, there can be at most k of those in a stable set. Thus, there cannot be more than $2n - k$ vertices in a stable set and $\alpha(G) \leq 2n - k$.

3.2 Question 2

First, it is clear that all the vertices out of the matching form a stable set or else we could have increased the number of edges in M. After coloring the vertices out of the matching, the only left are those covered by M. Note that they are also partitioned between U and V. When adding a new vertex u from the matching that is linked to a red vertex r in blue, we are sure that its matched vertex v is not yet coloured. Moreover, it cannot be adjacent to a red vertex. Indeed, if it were adjacent to b, we could have swapped the edge uv in the matching for the edges ru and bv and would have gotten a matching, which would enter in contradiction with the maximality of M. Thus, adding vertices as such preserves the fact that the red vertices form a stable set. Then, after this step, since two vertices in U are never connected by a single edge, we have the union of two independent sets, which gets us an independent set.
We now need to show that this set is maximal. Since that we first colour the $2n - 2k$ vertices that are not covered by the matching, and that then we add one by one half of the vertices (k) in the matching, we obtain a stable set of cardinal $2n - k$ and thus a maximum stable set.