22-2

(d)

根據 (b) 跟 (c)，對於所有點 u 檢查：

(1) DFS tree 的 root，若有兩個以上的 children 則是，反之則不是。

(2) DFS tree 的所有 leaves，都不可能是。

(3) 對於 u 來說，若存在一個 son s 且 $low[s] >= d[u]$ 的話， 則 u 是。

(意義: u 的 descendant 沒有人連到 u 的上面, 所以拔掉 u 會造成 disconnect)

Time ： $O(E)$ for DFS (because connected，$O(V+E) = O(E)$)

$O(E)$ for compute $low(u)$ for all vertex (see (c))

(1)(2)：$O(V)$

(3) each vertex will be checked at most twice，so total is $O(V)$.

Total : $O(V)$

(f)

若不是 tree edge，則不可能是 bridge。

根據 (e)，對於所有 tree edge (u,v) 檢查：

設 u 爲 v 的 parent，若 $low[v] = d[v]$ 的話， 則 (u,v) 是。

(意義: v 的 descendant 沒有連到 u 以上(包含 u) 的 edge, 則拔掉 (u,v) 會造成 disconnect)

Time ： $O(E)$ for DFS (because connected，$O(V+E) = O(E)$)

$O(E)$ for compute $low(u)$ for all vertex (see (c))

$O(1)$ time checking for each edge

Total : $O(E)$

23-1

(a) 證明 Minimum Spanning Tree 是 unique：

用反證法，先假設 MST 不是 unique，則至少存在兩棵 MST T_1 和 T_2。因爲 T_1 不等於 T_2，所以可以假設 T_1 有 k 條 edges 不包含在 T_2 之內，(k 至少>=1)，令 A 爲兩邊不同的 edges 所成集合，A 的 size 爲 $2k$。

令 A 中最小的 edge 爲 $e = (u,v)$，且不失一般性的假設 e 在 T_1 內。將 e 加入

T_2 會形成 cycle C，由於 T_1 不包含 C(因爲 T_1 是 tree)，所以 C 中必存在一條 edge f 不屬於 T_1，則 f 也是 A 中的一條 *edge*，因爲 edge weight 爲 distinct 且 e 和 f 都屬於 A，所以 $w(f) > w(e)$。

T_2 加上 e 減掉 f 會成爲新的 tree T_3，$w(T_3) = w(T_2) + w(e) - w(f) < w(T_2)$，$T_3$ 是更小的 spanning tree，違反 T_2 是 MST 的假設，矛盾。

故在 edge weight 爲 distinct 的情形下，MST 爲 unique。

證明 Second-Best Minimum Spanning Tree 不是 unique：

只要舉一個 graph 中有兩個（或以上）Second-best MST 的例子即可。

(b) 首先要證明任何 spanning tree S 只要不是 MST，就可以用一條相對於 S 的 nontree edge 和一條 S 的 tree edge 交換，得到總 weight 比 S 小的新 spanning tree S'：

令 T 爲那棵唯一的 MST，因爲 S 不等於 T，故必存在一條 edge $e = (u,v)$，使得 e 屬於 S 不屬於 T。將 e 從 S 取走會剩下兩個 component U 和 V，其中 U 包含 u 而 V 包含 v，而後再將 e 加入 T 會形成 cycle C，C-e 是一條由 u 通到 v 的 path，因此這條 path 一定要 cross U 和 V 兩邊，則 path 上必有一條 edge f，其兩端點分別屬於 U 和 V 兩個 component，因此 f 可以將 U 和 V 連起來，重新變成一棵新的 spanning tree S'。

又因爲 T 是 MST，所以 e 的 weight 必大於 C-e 上每一條 edge 的 weight，所以 $w(e) > w(f)$，故 $w(S') = w(S) - w(e) + w(f) < w(S)$，得證。

再來證明存在一對 edges (u,v) in T 和(x,y) not in T，使得 $T - (u,v) + (x,y)$爲 Second-best minimum spanning tree：

令 S 爲任意一棵 Second-best MST，由於 S 不是 MST(MST 爲 unique)，由上面的證明可知，存在一組 edge e 和 f，e 屬於 S，f 不屬於 S，使得 e 和 f 交換後得到 weight 更小的 spanning tree S'。

但是 S 爲 Second-best MST，故比 S 更好的 spanning tree 只有 T 而已，所以 $S'=T$。

令 $f = (u,v)$屬於 T 而 $e = (x,y)$不屬於 T，則 $T - f + e = T - (u,v) + (x,y) = S$，而 S 是 Second-best MST，得證。

(c) 在給定某一個 spanning tree T 的情形下，要計算所有 T 上的 $max[u,v]$，對 vertex u 而言，若要求出所有的 $max[u, v]$，只要以 u 爲 root，在 Tree T 上做 DFS，計算經過的每一個點與 u 的 max 即可。也就是說在 DFS 時，若由 w 對 v 做 DFS-visit，

則 $max[u,v]$ = edge with max-weight in $\{max[u, w], (w, v)\}$

若要求 All-pair 的 $max[u, v]$，則只要以每一點爲 root 做 DFS 即可。

Time : DFS 爲 $O(V + E) = O(V + V) = O(V)$ （因爲是 Tree）

做 V 次爲 $V * O(V) = O(V^2)$。

(d) 先利用 Prim's algorithm 計算出 MST T（在 c 是 given，在此不是），然後根據 (c)計算 max 之後，對每一條 non-tree edge $e' = (x', y')$，都計算加入 e'且去掉 $max[x', y']$的 spanning tree 其 weight 爲多少，並找出結果最好的 edge $e = (x, y)$。

則 Second-Best MST 就是 $T - max[x, y] + (x, y)$。

Time : 計算 MST T 需 $O(E + V \log V)$

計算 max function 須 $O(V^2)$

每一條 edge 的 test 爲 $O(1)$，所以時間總和爲 $O(E)$。

總共 $O(E + V \log V + V^2 + E) = O(V^2)$。