Centrale 2021

I Quelques Fonctions Auxiliaires

Q1.
```
let nombre_aretes g =
    let rec length_list l =
        match l with
        |[] -> 0
        |h::t -> 1 + length_list t
    in
    let s = ref 0 in
    for i = 0 to Array.length(g) - 1 do
        s := !s + length_list g.(i)
    done;
    !s / 2 ;;
```

Q2.
```
let g_2_3 = [|
    [|3; 1|];
    [|4; 0; 2|];
    [|5; 1|];
    [|0; 4|];
    [|1; 3 ; 5|];
    [|2; 4|];
    |];;
```

Q3.
```
let adjacence g =
    let n = Array.length g in
    let adj = Array.make_matrix n n 0 in
    for i = 0 to n - 1 do
    List.iter (fun x -> adj.(i).(x) <- 1) g.(i)
    done;
    adj ;;
```

Q4.
```
let rang (p, q) (s, t) =
    let is, js = s / p, s mod p in
    let it, jt = t / p, t mod p in
    if it = is + 1 then
        (q - 1) * js + is
    else if jt = js + 1 then
        p * (q - 1) + (p - 1) * is + js
    else
        failwith "Argument(s) invalide(s)" ;;
```

Q5.
```
let sommets (p, q) rg =
    if rg < p * (q - 1) then
        let is, js = rg mod (q - 1), rg / (q - 1) in
        let s = is * p + js in
        (s, s + p)
    else if rg < p * (q - 1) + q * (p - 1) then
        let shift = p * (q - 1) in
        let is, js = (rg - shift) mod (p - 1), (rg - shift) / (p - 1) in
        let s = js * (q + 1) + is in
        (s, s + 1)
    else failwith "Argument(s) invalide(s)" ;;
```

Q6.
```
let quadrillage p q =
    let graphe = Array.make (p * q) [] in
    let rec remplissage_graphe rg =
        if rg < p * (q - 1) + q * (p - 1) then
            let v1, v2 = sommets (p, q) rg in
            begin
```

```
            graphe.(v1) <- v2 :: graphe.(v1) ;
            graphe.(v2) <- v1 :: graphe.(v2);
            remplissage_graphe (rg + 1);
            end
    in
    remplissage_graphe 0 ;
    graphe ;;
```

II Caractérisation des arbres

II.A - Propriétés sur les arbres

Q7. Si $s, t \in S_n$, notons $s * t$, la relation "Il existe un chemin de s à t".
Montrons que $*$ est une relation d'équivalence sur S_n.

- Réflexivité : soit $s \in S_n$. Par convention, il existe un chemin de s à s. Donc $s * s$.
- Symétrie : soit $s, t \in S_n$, si $s * t$, alors il existe un chemin $c = (s, s1, \ldots, s_{k-1}, t)$. Donc $\forall i \in \{0, \ldots, k-1\}$, $\{s_i, s_{i+1}\} \in A$ donc $\{s_{i+1}, s_i\} \in A$, donc le chemin $c' = (t, s_{k-1}, \ldots, s_1, s)$ existe et donc $t * s$
- Transitivité : soit $s, t, u \in S_n$ tels que si $s * t$ et $t * u$. Alors il existe $c1 = (s, s_1, \ldots, s_{k-1}, t)$ et $c2 = (t, t_1, \ldots, t_{i-1}, u)$.
Donc en concaténant ces chemins, il existe $c = (s, \ldots, s_{k-1}, t, t_1, ;t_{k-1}, u)$, d'où $s * u$.

Ainsi comme les composantes connexes de G sont les classes d'équivalence de $*$ et forment donc une partition de S_n.

Q8. Soit s, t deux sommets tels que $s * t$, en notant $len(c)$ la longueur d'un chemin c, alors $L = \{len(c) | c \text{ chemin de } s \text{ à } t\}$ est une partie de $\mathbb{N}$, non-vide (puisque $s * t$), donc il existe un plus petit élément k_0 de L. D'où l'existence d'un plus court chemin de s à t.

Soit c_0 un plus court chemin de s à t, notons le $c_0 = (s, s_1, \ldots, s_{k_0-1}, t)$.
Si il existe $i \neq j$ tels que $s_i = s_j$ (on peut supposer sans perte de généralité que $i < j$) alors $c = (s, \ldots, s_i = s_j, s_{j+1}, \ldots, t)$, un chemin de longueur $k_0 - (j - i) < k_0$, ce qui contredit le caractère de plus court chemin de c_0 $\rightarrow$ absurde. Donc les sommets d'un plus court chemin sont distincts.

Q9. Soit $k \in [\![0, m[\![$, notons s, t les extrémités de a_k.

Supposons que s et t appartiennent à la même composante connexe de G_k, alors $s *_k t$. Ainsi en notant $c_k = (s_0 = s, s_1, \ldots, s_{i-1}, s_i = t)$ (avec $i > 1$), où les sommets de c_k sont adjacents dans G_k, alors il existe un chemin c dans G tel que $c = (s, \ldots, t, s)$ (car a_k relie s et t). Or $len(c) = i + 1 \geqslant 2$, donc il existe un cycle dans G. Or G est un arbre donc est acyclique $\rightarrow$ absurde !

Ainsi les extrémités de a_k appartiennent à deux composantes connexes différentes de G_k.

En notant pout tout $i \in [\![0, m]\!]$, $\varphi(i)$ le nombre de composantes connexes de G_i, alors $\phi(0) = n$ (G_0 est composé de n sommets non reliés) et $\varphi(m) = 1$ ($G_m = G$ est un arbre, donc connexe).

Donc si $k \in [\![0, m[\![$ et $a_k = \{s, t\}$, alors d'après ce qu'on a fait juste avant, s et t sont dans deux composantes connexes différentes de G_k et dans la même dans G_{k+1}. Les composantes connexes étant disjointes, si $u \in S_n$ tel que $C_{u_k} \neq C_{s_k}$ et $C_{u_k} \neq C_{t_k}$, alors $C_{u_k} = C_{u_{k+1}}$, (les C_{i_k} étant les composantes connexes de G_k contenant i), d'où finalement $\varphi(k+1) = \varphi(k) - 1$

Par une récurrence immédiate, $\varphi(m) = \varphi(0) - m$, d'où $m = n - 1$ et donc le résultat.

Q10. D'après **Q9.**, $(i) \implies (ii)$ et $(i) \implies (iii)$