

Algorithmique d

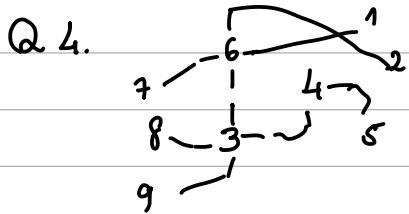
1D m° 5.

- Q1. (a) on fait un DFS sur $G(n)$
 (b) on calcule les degrés et on donne les sommets de degré 1:
 $\omega(G(n))$ $\omega(G(2n))$
- Q2. (a) On fait de la programmation dynamique : (on peut aussi procéder à un
 tri à l'aide d'un bucket, donn.)
- $$mwoe[x] := \min \left\{ \text{poids}(x) + \sum_{y \sim x} mwoe[y], \sum_{x \sim y} mwoe[y] \right\}.$$

(b) On fait un 1^{er} DFS pour trouver le sommet x le plus loin du sommet choisi arbitrairement. Ensuite, avec un 2nd DFS, on part de x et on regarde le sommet y le plus loin de x.

$$\text{diam}(T) = \text{profondeur de } y \text{ dans le 2nd parcours}$$

Q3. 6643633



On crée une file de priorité sur $[[1, |w|+2]]$ où les priorités sont les nb d'arc dans w, plus un.
 $i \leftarrow 0$

Solution: On barre des sommets. Tant que la file de prio n'est pas vide faire
 On lit le mot, à une lettre on la
 relie au plus petit qui n'est pas
 barré et qui n'est pas dans la
 séquence restante; puis on barre la
 lettre.

Extrire le min x.

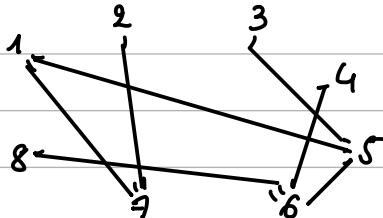
Relier x à w_i .

$i \leftarrow i+1$.

Retirer 1 à la prio de w_i et si
 $\text{d}_i \text{ prio} = 0$ alors on le retire.

Q5. $|T_n| = n^{n-2}$

756156



TD n° 6.

Q 1. (a) Soient T_1 et T_2 deux arbres courant de poids minimum.
 Supposons $T_1 \neq T_2$ d'où $E(T_1) \Delta E(T_2) \neq \emptyset$.
 Soit $e \in E(T_2) \Delta E(T_1)$ de poids min.

Sans perdre en généralité, supposons $e \in E(T_1)$.
 Le graphe $T_2 + e$ a un cycle C .

Soit $e' \in (C - \{e\}) \cap (E(T_2) \setminus E(T_1))$.

Alors, $T_2 + e' - e$ est un arbre courant de poids < poids de T_2 .
 Absurde.

Gm conclut $T_1 = T_2$.

(b) Soit $E = \{e_1, \dots, e_n\}$ tels que $w(e_1) \le \dots \le w(e_n)$.
 Gm pose $w'(e_i) := i$.

Comme les poids $(w'(e))_{e \in E}$ sont tous différents, alors
 on peut appliquer l'algorithme et avoir T .

Et, comme l'ordre défini par w' est un raffinement de l'ordre
 défini par w , on a que T est un ACPM pour w .

Q 2. Gm considère $G = (V, \delta_2(V), w)$ où $w(v, v') := d(v, v')$.
 Gm fait $n-k$ étapes de Kruskal.

Complexité en $O((n-k) \alpha(n))$.

Soit C le résultat d'espacement ε .

Soit C' un autre k -clustering.

Il existe u, v dans 2 composantes différentes de C' et dans la même composante de C .

Montre que $d(u, v) \le \varepsilon$ (ce qui implique espacement $(C') \le \varepsilon$).

Soit s tel que $d(s, t) = \varepsilon$.

Si $d(s, t) = \varepsilon < d(u, v)$ et on sait qu'et ne crée pas de cycle alors absurde car Kruskal aurait choisi et.

Q3. Arbres non enracinés

- Réflexivité : $\phi = \text{id}$

- Symétrie : $\phi' = \phi^{-1}$

- Transitivité : $\phi'' = \phi' \circ \phi$

$$V_1 \xrightarrow{\phi} V_2 \xrightarrow{\phi'} V_3$$

$\underbrace{\quad}_{\phi'' = \phi' \circ \phi}$

Arbre enraciné :

- Réflexivité : $\phi = \text{id}$

- Symétrie : $\phi' = \phi^{-1}$

- Transitivité : $\phi'' = \phi' \circ \phi$

$$\pi_1 \xleftarrow{\phi} \pi_2 \xleftarrow{\phi'} \pi_3$$

$\underbrace{\quad}_{\phi'' = \phi' \circ \phi}$

Q4.

$A \sim D$ avec c l'isomorphisme

$$\begin{array}{llll}
 a & w & r & u \\
 b & z & g & s \\
 c & y & h & t \\
 d & x & i & v \\
 e & v & j & pu
 \end{array}$$

$C \not\sim A, D$ car il existe un sommet de degré 4 relié à trois feuilles dans C mais pas dans A.

$B \not\sim A, C, D$ car $\deg_B(2) = 2$ et $\deg_A(\cdot) \neq 2$ $\deg_C(\cdot) \neq 2$.

Q5. $\forall x \in V(T-F), R_{T-F}(x) = R_T(x)-1$

$$\begin{aligned}
 C(T-F) &= \{x \in V(T-F) \mid R_{T-F}(x) = R(T-F)\} \\
 &= \{x \in V(T-F) \mid R_T(x) = R(T)\} \\
 &= C(T)
 \end{aligned}$$

Q6. Par récurrence forte sur $\#T$. Complexité en $\mathcal{O}(n)$, c.f. TD 5.

Q7 $T \sim T' \iff \exists \pi \in C(T), \exists \pi' \in C(T'), (\tau, \pi) \sim (\tau', \pi')$

" \Leftarrow " oui

$$\Rightarrow R(\phi(x)) = R(x)$$

$$\text{d'où } C(T) = C(\phi(T)) = \phi(C(T)).$$

Q8. On calcule $C(T)$ et $C(T')$ en $\mathcal{O}(n)$.

Soit $x \in C(T)$.

Pour tout $x' \in C(T')$, tester $(\tau, x) \sim (\tau', x')$.

Complexité en $\mathcal{O}(n \cdot 2^{\#f(n)}) = \mathcal{O}(f(n) + n)$.

TD n° 7

I Graphes bipartis

Q1. Si G est biparti et qu'il a un $(2k+1)$ -cycle alors



Absurde car $X \neq Y$.

Réiproquement, si G n'est

Q2. DFS en $O(n+m)$ pour avoir un 2-coloriage

II Tri topologique par élagage

Q3. On part que $u \in V$.

Tant que $\deg^+(u) > 0$ faire
 $u \leftarrow$ un préédécesseur de u

Q4. cycle $\Rightarrow x_1 < \dots < x_n$ dans le tri topo

$x_1 \rightarrow \dots \rightarrow x_n \rightarrow x_1$ $< x_1$ absurde.

acyclique \Rightarrow tri topo

Soit v de $\deg^+(v) = 0$.

$v \in \boxed{\text{tri topo de } G - v}$

↑ acyclique

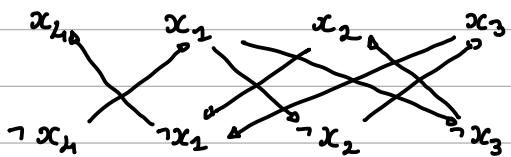
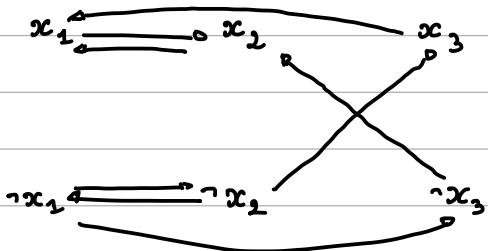
Q5. On calcule tous les \deg^+ que l'on maintient.

On extrait tous les sommets de $\deg^+ = 0$ (dans une pile)

III Graphes pour 2-SAT

$$\neg p \vee q = p \rightarrow q$$

Q6



Q7. Que le graphe ne contienne pas de cycle avec x_i et $\neg x_i$.

$$f: \mathcal{X} \rightarrow \mathbb{B}$$

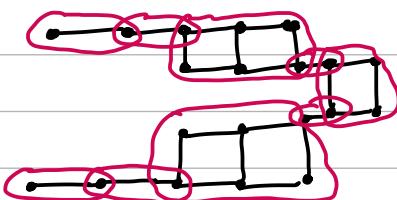
$$x_i \leftarrow \text{Vrai si } x_i < \neg x_i \\ \downarrow \\ \text{tri topo}$$

IV Points d'articulations, ponts, et composantes 2-commexes.

Q9. Points d'articulation : II, IX, XII, XIX

Ponts : II-III, VIII-IX, XIII-XII, XIX-XVIII

Comp. bi-commexes :



Q10. Si r est un point d'articulation avec < 2 fils alors rebire r ne déconnecte pas le graphe. Absurde!

Si la racine $r \geq 2$ fils alors les sous arbres des fils de r sont des composantes commexes de $G-r$.

D'où point d'articulation.

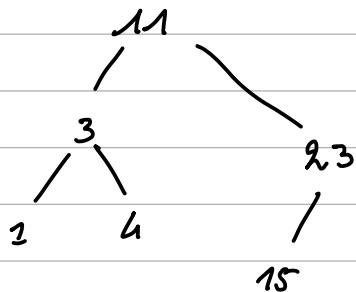
Q11.

10_{n° 8}

I. Jeu des erreurs

Q1. Vrai

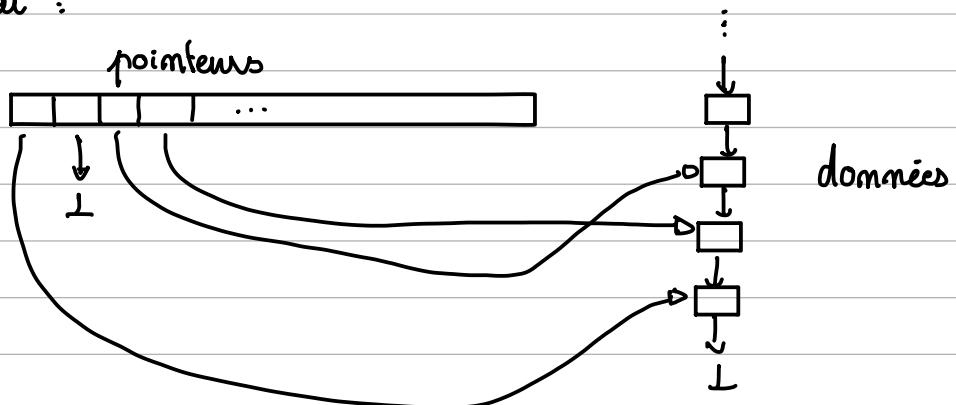
Q2. Faux : $15 < 6$



Q3. Faux : 20 18 il faut faire de 7 à 1

Q4. Faux : pas d'hypothèse sur le poids de e

Q5. Vrai :



Q6. On fait l'algorithme de Prim *jaunié* avec trois buckets

U

arêtes

de poids 1

U

arêtes

de poids 2

U

arêtes

de poids 3

on peut remplir
ces arêtes par DFS
du graphe

II Coloriage minimal et k -ième minimum

Q7. Par programmation dynamique : $\text{opt} : V \times [1, \Delta+1] \rightarrow \mathbb{N}$
 où $\Delta = \max_{v \in V} \deg(v)$

$$\text{opt}(u, i) = i + \sum_{\substack{v \in V \\ u \rightarrow v}} \min_{1 \leq j \leq \deg(v)+1} \text{opt}(v, j)$$

Algo en $\sum_{u \in V} \sum_{\substack{v \in V \\ u \rightarrow v}} d(v)$
 $\underbrace{\sum}_{\leq |E|} \underbrace{\sum}_{\leq \Delta}$

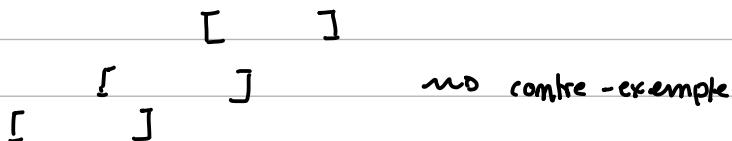
Q8. k as & k extractions de min

ou... quick select

III Graphes dynamiques

Q9. (a) oui

$$(b) \Rightarrow \text{oui} \iff c_i = \max(t_{i,i}, t_{i+1})$$



(c)

$$\text{TCM} := \min_{\substack{\text{chemins} \\ \text{de contagion}}} c_l$$

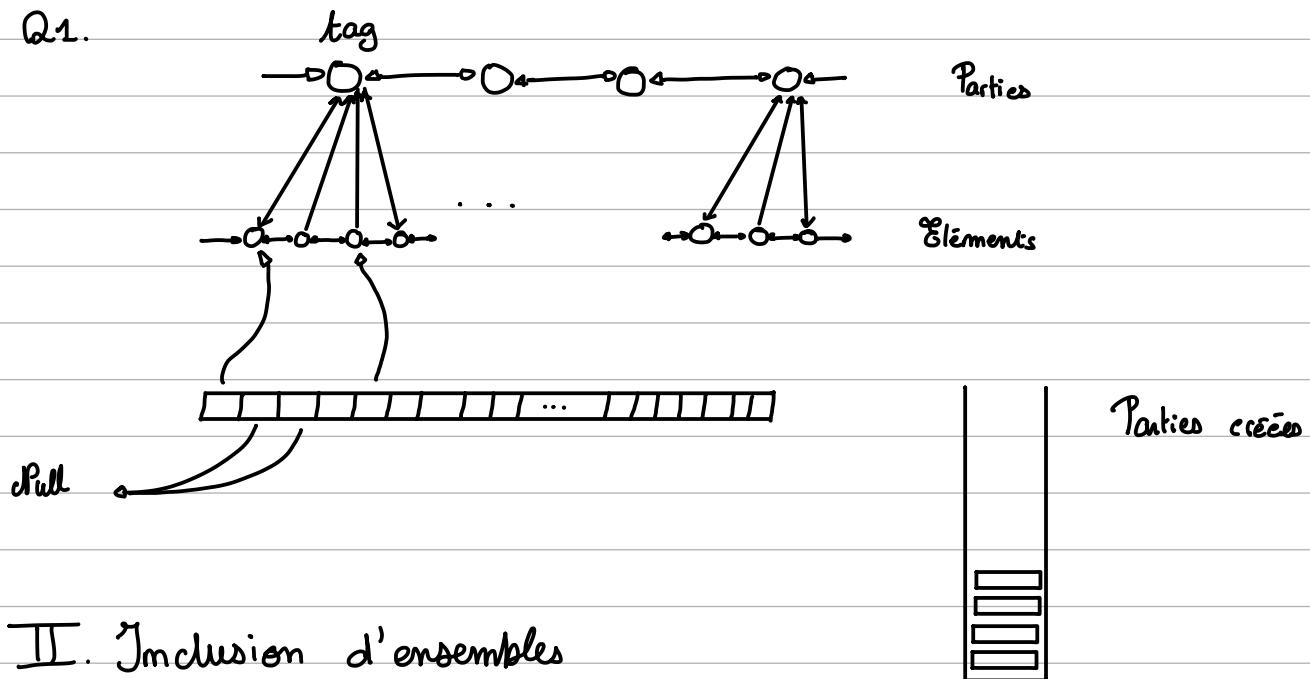
parcours du graphe en $O(|V| + |E|)$

Q10. On fait un parcours avec une file de priorité.

TD n° 9

I. Raffinement de partition

Q1.



II. Inclusion d'ensembles

Q2. (a) On construit une table de hachage.

On ajoute (i, x) à la table pour tout i et tout $x \in A_i$.

On teste si $(T[i], x)$ est dans la table pour tout i , tout $x \in A_i$.

\hookrightarrow oui pour tout $i, x \Rightarrow$ oui

\hookrightarrow non pour un certain $i, x \Rightarrow$ non

D'où une complexité en $O^k(n+m)$

(b) Solution magie magie

$$\left. \begin{array}{l} M \leftarrow \text{malloc } (n^2) \\ V \leftarrow \text{malloc } (n^2) \end{array} \right\} \begin{array}{l} \text{magie magie} \\ \text{c'est un } O(1) \end{array}$$

cnt $\leftarrow 0$

Add $((i, x)) :=$

$$\left[\begin{array}{l} M[i \cdot n + x] \leftarrow \text{cnt} \\ V[\text{cnt}] \leftarrow i \cdot n + x \\ \text{cnt} \leftarrow \text{cnt} + 1 \end{array} \right]$$

Membership $((i, x)) :=$

$$\left[\begin{array}{l} \text{Si } M[i \cdot n + x] \geq \text{cnt} \text{ alors faux} \\ \text{Sinon si } V[M[i \cdot n + x]] \neq i \cdot n + x \\ \quad \quad \quad \text{alors faux} \\ \quad \quad \quad \text{sinon vrai} \end{array} \right]$$

D'où une complexité en $O(n+m)$

Autre solution :

Pour tout j , $U[j] \leftarrow \{i \mid T[i] = j\}$

$E \leftarrow [\text{faux}, \dots, \text{faux}]$

Pour tout j

Pour tout $x \in A_j$, $E[x] \leftarrow \text{vrai}$

Pour tout $x \in U[j]$

[Pour tout $y \in A_i$: Tester $E[y]$]

Pour tout $x \in A_j$, $E[x] \leftarrow \text{faux}$

III LexBFS

Q3. Si $L[j] \rightarrow L[i] \rightarrow L[k]$. où $i < j < k$
alors

si, au moment de traiter l'indice i , $L[j]$ et $L[k]$ sont dans la même partie alors
on a $k < j$, ce qui n'est pas possible.

D'où $L[j]$ et $L[k]$ sont déjà dans des parties différentes.

Soit t minimal tel que $L[i] - L[j] \neq L[t] - L[k]$.

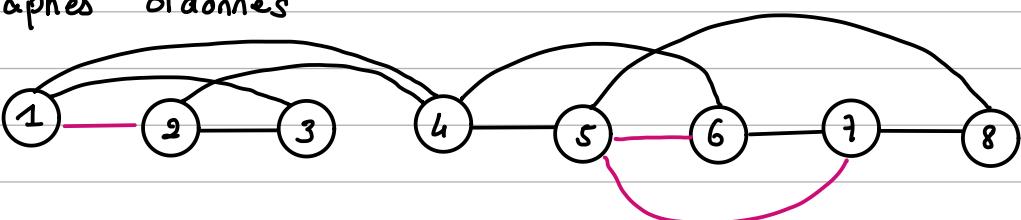
Par $j < k$ alors $L[t] - L[j]$ et $L[t] \rightarrow L[k]$.

Q4. (a) $\sum_{x \in V} (2 \cdot G(1) + G(\#N(x))) + G(\#V) = G(2n + 2m)$

(b) On extrait le max avec la structure de Q1.

IV Graphes ordonnés

Q5.



Q6. Si G' simplicial et $G^*(G') = G$, $G \leq G'$
alors

$$G \subseteq G^*(G) \subseteq G^*(G') \subseteq G$$

d'où G simplicial

Q7.

(a) On vérifie $\underbrace{N^-(i)}_{A_i} \subseteq \underbrace{N^-(j) \cup \{j\}}_{A_{j+m}}$ pour tout i et $j = \min N^-(i)$.

Complexité en $O(n+m)$.

(b) $\llbracket 1, i-1 \rrbracket \setminus N^-(i) \subseteq (\llbracket 1, j-1 \rrbracket \setminus N^-(j)) \cup \{j\}$
ssi $\llbracket 1, i-1 \rrbracket \setminus N^-(i) \subseteq \llbracket 1, i-1 \rrbracket \setminus N^-(j)$
ssi $N^-(i) \supseteq N^-(j)$

On peut tester ça pour tout i , où $j = \max \{i + j'\}$