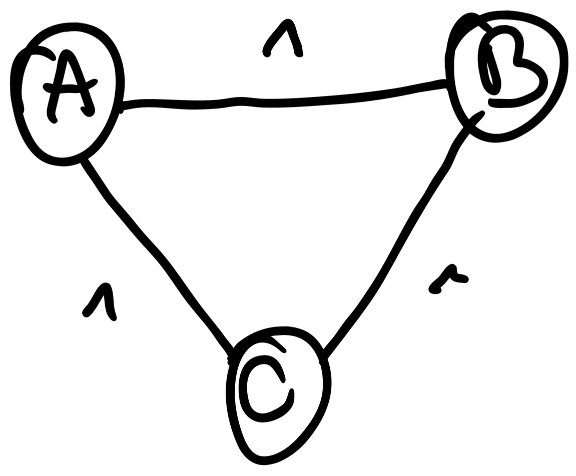
-TD4 - suite

3. 

⤳ On va sélectionner les 3 arêtes ce n'est donc pas un acm.

↳ si on choisit juste une arête de poids minimal

| A choisit (A,B)
| B choisit (B,C)
| C choisit (C,A)

Deux solutions ;

① dans l'algo on choisit l'arête avec le sommet minimal

↳ | il faut ordonner les sommets
| (ex : ordre alphabétique)

⤳ A choisit B
B choisit A
C choisit A.

② rendre tous les poids différents

↳ | ex : 1 et 1
| ⇓
| 1 et 1,000..01

- TD 10 -

Exercice 1 :

Preuve : Supposons que cette stratégie donne les arrêts $b_1, b_2 \ldots b_n$
Supposons qu'il y a une suite d'arrêts $a_1, a_2, a_3 \ldots a_k$ avec $k < n$
On montre que ce n'est pas possible :
on prend la première différence entre les 2 suites

glouton $b_1 \quad b_2 \ldots b_i \quad b_{i+1} \ldots$
$\quad\quad\quad\; \shortparallel \quad \shortparallel \quad\; \shortparallel \quad \neq$
l'autre $a_1 \quad a_2 \ldots a_i \quad a_{i+1} \ldots$

$\underbrace{a_{i+1} \text{ est à vant } b_{i+1}}$?
on remplace a_{i+1} par b_{i+1} et on a toujours une solution optimale plus proche de la solution gloutonne

Exercice 3 :

$$1 \xrightarrow{+1} 2 \xrightarrow{+1} 3 \xrightarrow{\times 2} 6 \xrightarrow{\times 2} 12 \qquad \text{4 opérations}$$

1er algo glouton : on double dès que possible sinon on fait +1. ⟶ pas optimal car :
$$1 \xrightarrow{\times 2} 2 \xrightarrow{\times 2} 4 \xrightarrow{\times 2} 8 \xrightarrow{+1} 9 \xrightarrow{+1} 10 \xrightarrow{+1} 11 \xrightarrow{+1} 12$$

2ème algo glouton : on part de la cible et on construit la suite à l'envers :
- si la cible est paire on divise par 2
- sinon on enlève 1

Preuve :
- si la cible est impaire, la dernière opération doit être +1
- si la cible est paire
  Supposons que c'est +1 ($\neq$ de ce que fait l'algo2)
  On considère le dernier ×2

(si il n'y en a pas l'algo 2 est meilleur)

$K \xrightarrow{\times 2} 2k \xrightarrow{+1} \dots \xrightarrow{+1} 2m = n$ ⇝ $n - 2k + 1$ operations

On peut remarquer :

$K \xrightarrow{+1} K+1 \xrightarrow{\times 2} 2k+2 \xrightarrow{+1} \dots \xrightarrow{+1} 2m = n$ ⇝ $n - 2K$ operations

⇝ nouvelle sol meilleur.

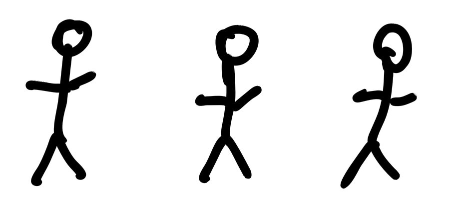
généralisation

operations : $+1$, $\times 2$, ~~$\times 5$~~, $\times 3$

$1 \xrightarrow{\times 3} 3 \xrightarrow{+1} 4 \xrightarrow{+1} 5 \xrightarrow{\times 2} 10$

$1 \xrightarrow{\times 3} 3 \xrightarrow{\times 3} 9 \xrightarrow{+1} 10$ ⇝ contradiction car avec l'algo trouvé on aurait pas le bon nombre d'operations

Exercice 4 :



1.7	1.9	1.8		1.7	1,8	1.9
"	"	"	⇝	"	"	"
2.1	1.8	1.6		1.6	1.8	2.1

Algo : On donne les skis les plus courts à la personne la plus petite et ainsi de suite

Optimalité

Si l'algo n'est pas optimal il existe une exemple où l'algorithme ne donne pas la même solution : t_1, t_n et $S_1, \dots, S_n$

Soit $(t_1, S_{\sigma(1)}) \dots (t_n, S_{\sigma(n)})$
la solution optimale s

Soit $(t_1, S_1) \dots (t_n, S_n)$ le résultat du glouton :

Soit t_i le premier skie où ça diffère

t_i $S_{\sigma(i)}$ Soit $S_j = S_{\sigma(i)}$ le ski assigné à i

l'ordre croissant

$S = (t_1, s_1)(t_2, s_2)(t_3, s_3) \dots (t_{i-1}, s_{i-1})(t_i, \underline{s_j}) \quad (t_k, s_i)$

$\cancel{(t_1, s_1)}(t_2, s_2)(t_3, s_3) \dots (t_{i-1}, s_{i-1})(t_i, \underline{s_i})$

on obtient un ~~S' avec~~

$(t_1, s_1)(t_2, s_2)\cancel{(t_3, s_3)} \dots (t_{i-1}, s_{i-1})(t_i, s_i) \dots (t_k, s_j)$

Il faut montrer que S reste optimale

$\text{Cout}(S) - \text{Cout}(S') = |t_i - s_j| + |\cancel{t_k} - s_i| - |t_i - s_i| - |t_k - s_j|$

Il faut montrer que $\text{cout}(S) - \text{cout}(S') \geq 0$ $\quad i = \sigma(k)$

On sait que $s_i \leq s_j \quad t_i \leq t_k$

Analyse par cas :

1. $s_i \leq s_j \leq t_i \leq t_k$
2. $s_i \leq t_i \leq s_j \leq t_k$
3. $s_i \leq t_i \leq t_k \leq s_j$
4. $t_i \leq s_i \leq t_k \leq s_j$
5. $t_i \leq t_k \leq s_i \leq s_j$
6. $t_i \leq s_i \leq s_j \leq t_k$

Il faudrait prouver tous les cas ci-dessus. Mais ici on va que prouver le 2.

on montre Prop pour 2 :

$|t_i - s_j| + |t_k - s_i| + |t_i - s_i| + |t_k - s_j|$

$= -t_i + s_j + t_k - s_i + t_i - s_i + t_k - s_j$

$= 2t_k - 2s_i \geq 0$

On considère un autre algo :

On prend i, j tq $|t_j - S_i|$ est minimal et on assigne à j le ski i

et continue avec les skieurs et skis qui reste avec le même principe.

~~Est-ce que cet algo est optimal ?~~

↳ Contre exemple : $t_1 = 5$ $t_2 = 10$

$S_1 = 1$ $S_2 = 6$