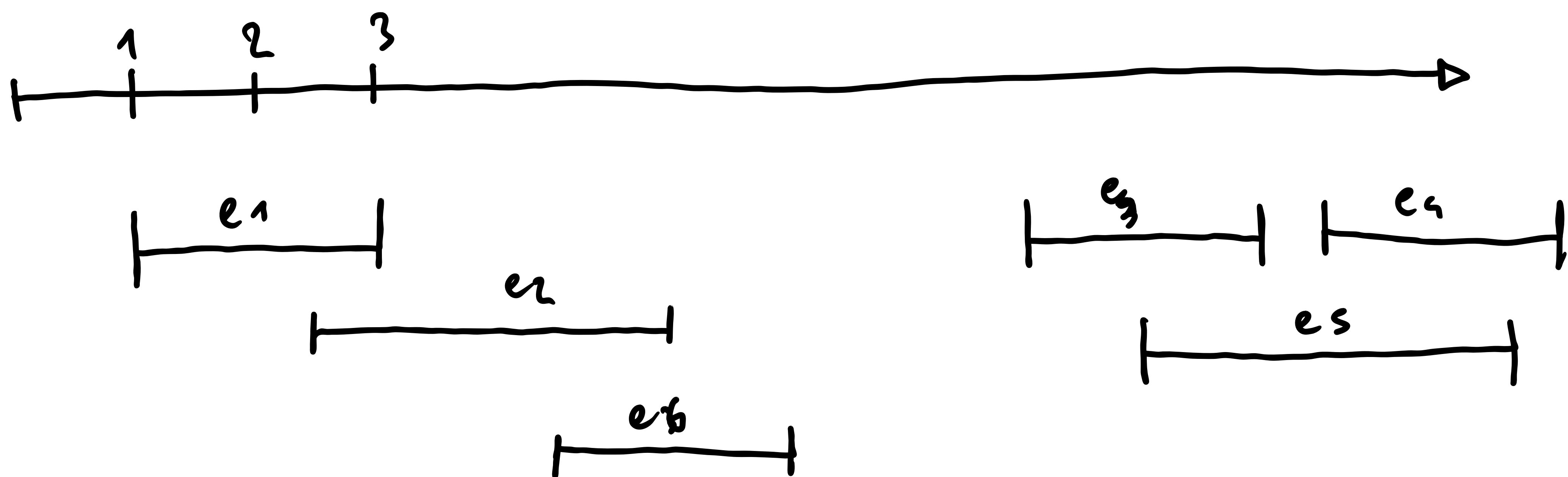


Algo gloton faire des choix locaux

↳ vérifier si la manière de choisir local est opti

ex prob location voiture



⇒ Sol: trier par ordre croissant

Def:

1) trier ordre req par ordre \uparrow $O(n \cdot \log n)$

2) $S := \emptyset$

3) Pour chaque req (d, h) $O(n)$

↳ si (d, h) est compatible avec S , on l'ajoute

4) retourner S

↳ date de fin / début ne se chevauchent pas

⇒ $O(n \cdot \log n)$

Correction:

Preuve 1:

Supposons que F_{gloton} n'est pas optimal

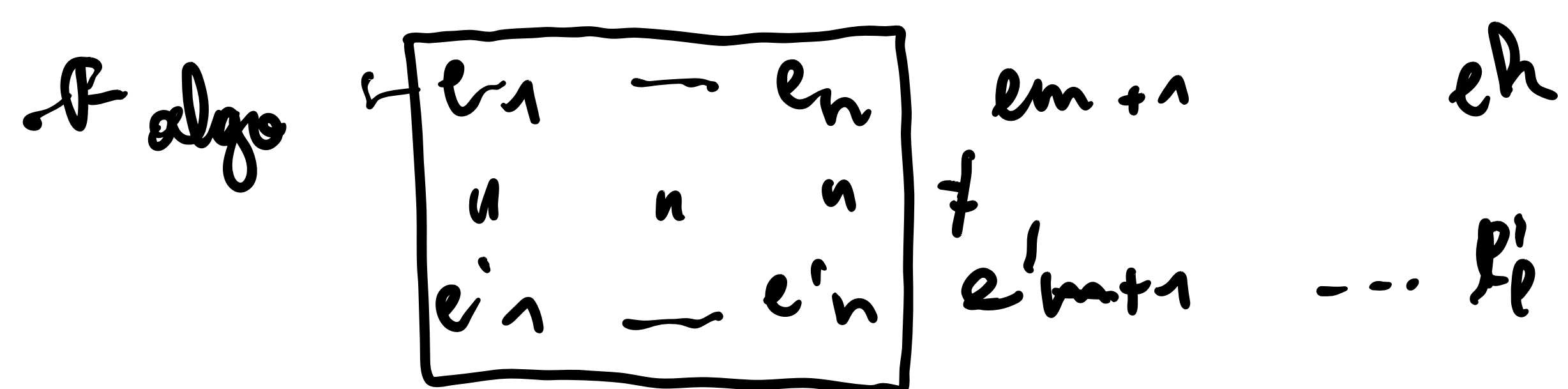
Soit S une solution optimale $|S| > |F|$.

$|S| = l > k$

avec $S = \{e'_1, e'_2, \dots, e'_l\}$

on choisit une solution optimale S qui partage le + de "premiers choix" faits par l'algo:

Cad : e_1, e_2, \dots, e_m



on sait que e_{m+1} est une req compatible
 avec $e_1 \dots e_m$ ayant une date de fin
 minimale

$$\Rightarrow h(e_{m+1}) \leq h(e'_m)$$

Preuve 2:

• Proposition 1: "bon premier choix"

Soit E un ens. de requêtes,

soit $e = (d, h)$ une req de E ayant une date de fin minimale,

alors : \exists une solution contenant e .

Preuve : soit S une solution

si $e \in S$ ok

si non on trie S par date de fin \hookrightarrow

$$S = \{(d_1, h_1), \dots, (d_n, h_n)\}$$

$$e = (d, h) \text{ est } h \leq h_1$$

$\rightarrow S \setminus \{(d_1, h_1)\} \cup \{(d, h)\}$ est une solution

• Proposition 2

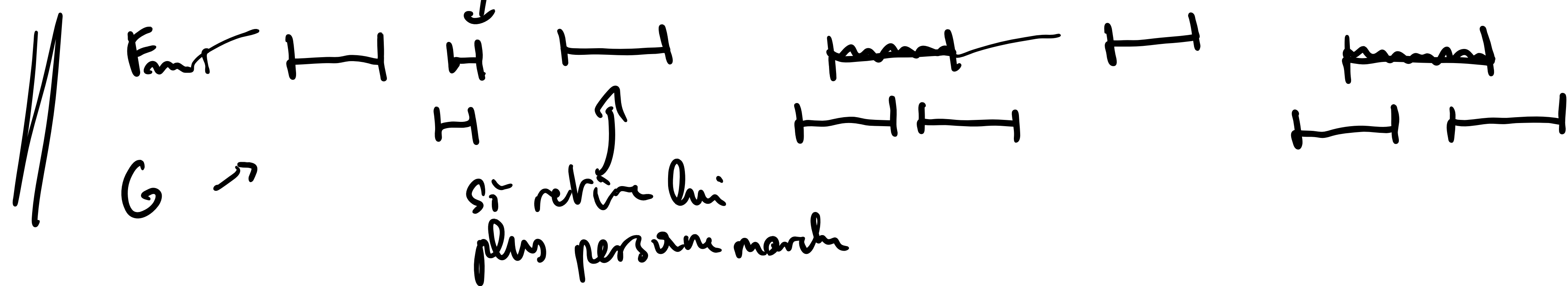
Soit E un ens. de n req

soit G une solution pour E ,

Soit $e \in G$

Alors :

$G' = G \setminus \{e\}$ est une solution pour l'ens des req de E compatibles avec e .
 si lui est retiré, les sol marchent si non compatible avec celle où lui est inclus



rq qui était uniquement dans F sinon dans $F \cup G$.

On mélange tout ça :

inductie sur $|E|$

$|E| = 1$: la sol de E

$|E| > 1$: soit S sol opt pour E

on sait (prop 1) que \exists une contenance la première $rq(d_1, h_1)$

choisie par l'algo à la 1^{re} itération

$S \setminus \{(d_1, h_1)\}$ est une solution pour les requêtes de E comp avec (d_1, h_1)

Par induction, mon algo est correct pour les cas de requêtes $< |E|$

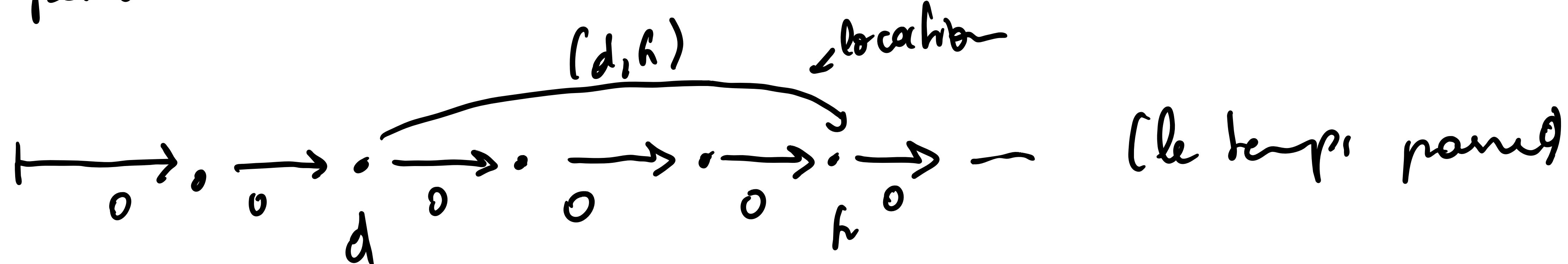
c'est le cas pour les requêtes de E compatibles avec (d_1, h_1)

\Rightarrow l'algo renvoie pour " E compatibles avec (d_1, h_1) " une sol de taille opt $\Rightarrow |S| - 1$

La sol complète contient donc $|S| - 1 + 1$ $rq = |S| \rightarrow$ optimale.

le plus de personnes doivent louer

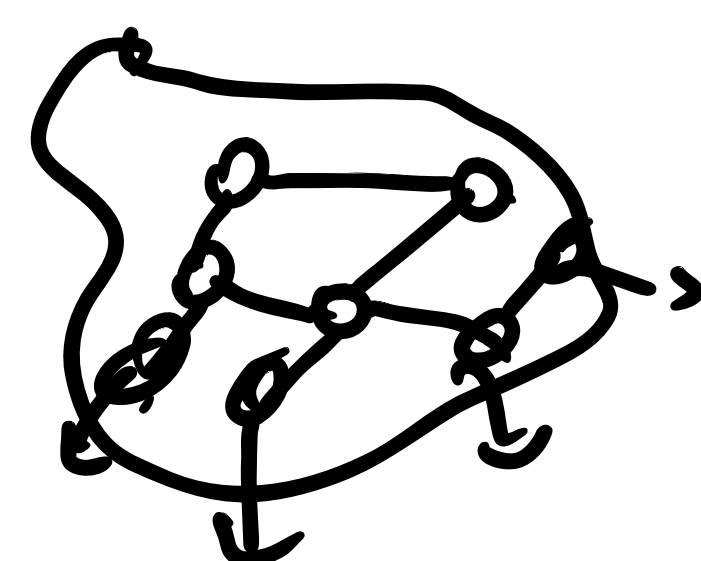
les personnes les + rentables doivent être (maximiser durée location)



Rq: possible faire prob sac à dos de avec algo gloton

\hookrightarrow faire 'ratio' entre valeur et poids

Algo dijkstra (PCC) pas poids neg car donne au fur et à mesure graph



Formules de Horn 3-CNF

Algo:

• $v = \{1, \dots, 2\}$

tant qu'il y a implémente mon sat, mettez la var à droite de " \Rightarrow " à T

vérifier que tt les clauses sont satisfaites

si oui: la formule est SAT

sinon: la form n'est pas SAT

—
ACM: — Prim (traitement en abr)

— Kruskal / trier puis les prendre dans l'ordre ↗

↳ Union-Find → analyse amortie.