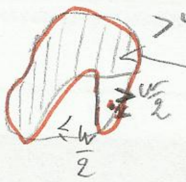
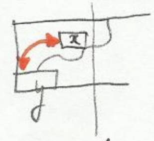
lemme 1 : I faisable sur $i \dots m$ (en w) $\Rightarrow I'$ faisable sur $i+1 \dots m$ (en w).

PROOF: $I =$  fit pas Soit $\tilde{I} = I' \cup \{x\} = I \setminus I_{small}$.

I faisable sur $i \dots m \Rightarrow \tilde{I}$ faisable sur $i \dots m$.

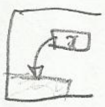
$\exists$ solution S^* de $\tilde{I}$: 

Si S^* a une tâche $y \in Big_i$ sur machine i.   (on swappe pr revenir à la m̶ instance que nous)

↳ si $y = x$, gagné, S^* restreinte à $i+1 \dots m$ est une solution de I'.

↳ sinon, swappe x et y.

La solution (après SWAP) restreinte à $i+1 \dots m$ ~~restreinte~~ nous prouve que I' est faisable sur $i+1 \dots m$.

Si S^* n'a pas de $y \in Big_i$ schédulé sur $i \Rightarrow$  $\square$

lemme 2 : $\forall i$, $A(w, I, i)$ fail $\Rightarrow I$ pas faisable sur $i \dots m$.

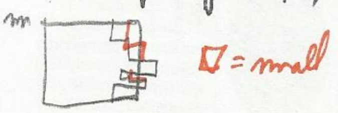
PROOF: → pour $i = m$, ok

→ i sachant $i+1$. Si $A(w, I, i)$ fail. Si c'est l'appel récursif qui fail, alors avec lemme précédent + réc, c'est ok.

Si c'est le remplissage greedy qui FAIL, alors $W(I) = \sum_{j \in I} p_j > w \left(\sum_{l=1}^{m} s_l \right)$

donc $OPT(I) \geq \frac{W(I)}{\sum_{l=1}^{m} s_l}$ $\square$

lemme 2.1 : greedy (w, i, I_{small_i}) : tant que $\exists$ machine l, $i \leq l \leq m$ qui finit avant w.



→ si reste tâches, FAIL ajouter x ($\in I_{small_i}$) sur l.

lemme 3 : Si pas fail, $A(w, I, i) \leq \frac{3w}{2}$.

PROOF: le seul endroit où l'on peut dépasser w est dans greedy.

Or $j \in small_i \Rightarrow j \in small_l$ $l \geq i$