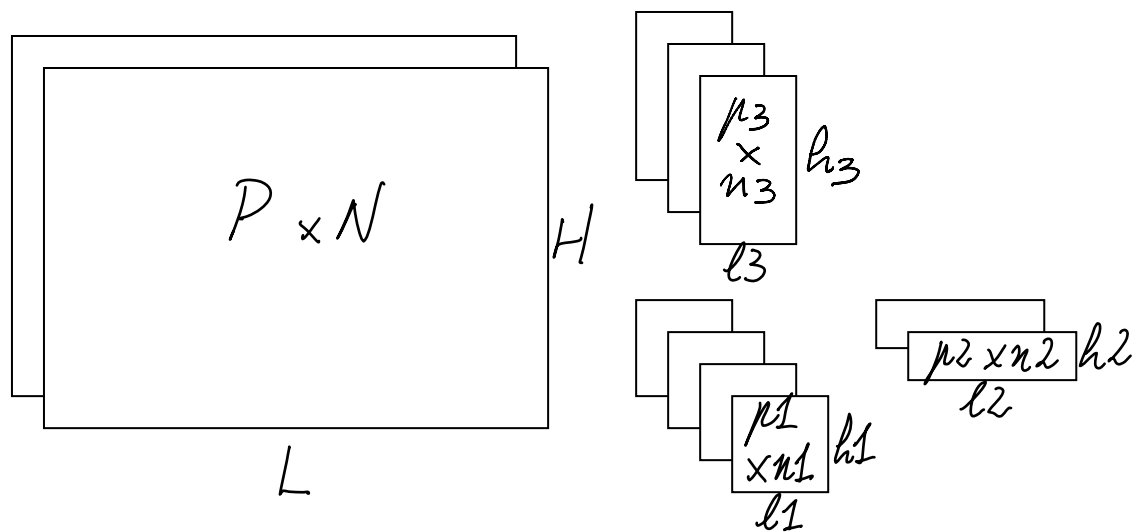


L'entreprise de tôlerie Elaa est spécialisée dans la découpe de tôles rectangulaires sur mesure. A jour J-1, elle doit planifier les coupes demandées jusqu'à ce jour J-1 par ses clients pour le jour J. Pour ce faire, elle dispose d'une machine M de découpe de haute précision qui est capable de couper dans une grande plaque rectangulaire P de tôle de dimensions (H,L) des pièces de tailles plus petites, toujours rectangulaires, p_i de dimensions (h_i, l_i). H et h_i représentent la hauteur des pièces, L et l_i leur longueur. L'entreprise dispose de N plaques P et chaque type de pièce p_i est commandé n_i fois. Par ailleurs, à cause des motifs de la plaque de tôle et des problèmes de résistances selon le sens de la coupe, on n'autorise aucune rotation des pièces p_i .



Le patron d'Elaa fait donc appel à vous, jeune ingénieur maîtrisant les nouvelles technologies, pour concevoir ces plans de coupe. Ici, il s'agit d'un problème de découpe rectangulaire où il faut placer les pièces p_i sur les grandes pièces P au mieux de manière à découper toutes les pièces p_i lorsque cela est possible et de minimiser les chutes (dimension des pièces P utilisées moins celles des pièces p_i placées).

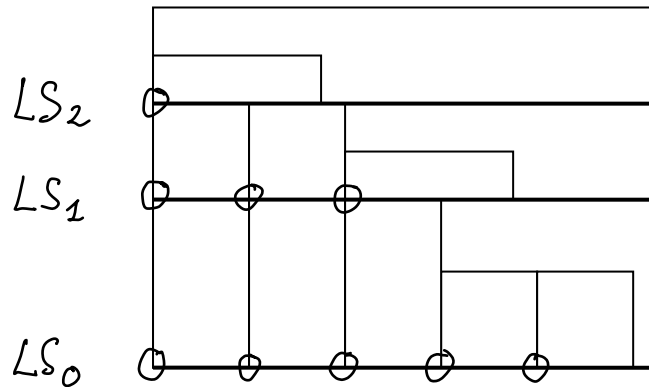
Après une étude bibliographique du domaine, il existe une méthode constructive de plan simple, connue dans la littérature sous le nom de *Bottom Left* (BL). Cette méthode s'exprime comme suit :

1. Prendre une nouvelle plaque P tant qu'il en reste.
2. Calculer la hauteur d'une ligne (LS) « support » courante ($LS_0=0$). Les lignes « support » sont représentées par des traits épais dans la figure exemple ci-après.
3. Prendre la plus grande des pièces p_i selon le critère de la surface $h_i \times l_i$.
4. Placer la pièce p_i sur la ligne « support » courante, en commençant toujours par le coin le plus bas et à gauche (*Bottom Left*) et en vérifiant que cette pièce p_i respecte les contraintes de dimensions. La position d'une pièce p_i sur une grande plaque P est donnée par celle du coin inférieur gauche de la pièce.
Si la pièce p_i ne respecte pas les contraintes de dimensions, prenez la pièce p_j immédiatement plus petite que p_i selon la surface et recommencez l'étape 4.
5. Répéter les étapes 3 et 4 pour remplir une ligne « support ». Lorsqu'une ligne « support » est pleine, on va à l'étape 2 en calculant la hauteur de la nouvelle ligne courante LS_i par la formule $LS_i=LS_{i-1}+\max(h_i)$, i.e. la hauteur de la nouvelle ligne de « support » i est égale à

celle de la ligne $i-1$ augmentée de la plus grande hauteur des pièces p_i placées sur la ligne $i-1$.

6. Lorsqu'une plaque P est pleine, on va à l'étape 1.
7. Cette méthode s'arrête lorsque soit toutes les pièces p_i sont déjà coupées, soit plus aucune pièce p_i ne peut être coupée pour cause d'insuffisance de plaque P ou pour cause de contraintes de dimensions.

La figure exemple ci-après donne un plan de coupe que cette méthode BL peut obtenir.



Les petits cercles donnent les coins inférieurs gauches permettant de repérer la position des pièces p_i .

Les données du problème sont fournies sous la forme d'un **fichier de données** suivant :

```
1 3
40 55 2
10 10 4
5 18 2
20 10 3
```

La 1^{ère} ligne donne le nombre de types de plaque et de pièces. Ici, nous ne considérons qu'un seul type de plaque P et n types de pièces p_i avec $n=3$ ici.

Chaque ligne donne les dimensions et le nombre de la plaque P (2^{ème} ligne) et de chaque type de pièce p_i (les lignes suivantes).

Le plan de coupe de la figure exemple s'écrit dans un **fichier de résultats** comme suit :

Plaque 0 :

LS=0

0 20 10, 10 20 10, 20 20 10, 30 10 10, 40 10 10,

LS=20

0 10 10, 10 10 10, 20 5 18,

LS=30

0 5 18,

Plaque 1 :

Pas utilisée.

Pièces restantes à couper :

Aucune.

Chutes:

1020

On donne la configuration de chaque plaque P et les pièces restantes à couper (s'il y en a).

Pour chaque plaque P, on donne toutes hauteurs des lignes « support ».

Pour chaque ligne « support », on donne les pièces sous forme de triplet. Chaque triplet correspondant donc à l'abscisse sur la ligne « support » et aux dimensions d'une pièce placée.

Pour les pièces restantes, s'il y en a, il est demandé de donner chaque type de pièces p_i également sous forme de triplet $nr_i \quad h_i \quad l_i$, où nr_i est le nombre de pièces de type p_i de dimensions (h_i, l_i) restantes à couper.

Sur notre exemple, la 2nde plaque P n'est pas utilisée et toutes les pièces p_i ont été coupées sur la 1^{ère} plaque P.

Tâches à effectuer dans ce projet :

Tâche 1 : l'écriture d'un algorithme et son programme JAVA associé pour la génération de plan de coupe fondé sur la méthode BL.

1. Faire une analyse descendante du problème au niveau 0:

- **Pour les données** : Identifiez toutes les classes d'objets ainsi que tous les attributs de chaque classe permettant de représenter les données que vous avez à manipuler pour créer un plan de découpe, et dressez un diagramme de relations entre ces classes (à la UML). Si des structures de données sont employées, précisez et justifiez-les.
- **Pour les traitements** : Donnez pour chacune des classes, les noms des méthodes, leurs entrées et sorties, et les grandes étapes de chacune des méthodes. N'hésitez pas à faire des schémas pour expliquer ce que vous voulez faire. Un schéma simple vaut parfois mille lignes d'explication.

2. Préciser votre analyse descendante au niveau i : pour chacune des méthodes, détaillez algorithmiquement les traitements. Si une étape de votre méthode est trop complexe n'hésitez pas à la décomposer en de nouveaux sous-algorithmes que vous préciserez par la suite et séparément, et ainsi de suite.

Tâche 2 : développer un algorithme et son programme JAVA associé qui améliore celui de la méthode BL.

Tâche 3 : dans le cas où on a une seule plaque grand P, développer un algorithme exact et son programme JAVA associé qui trouve la meilleure solution en plaçant un maximum de petites pièces p_i .

Tâche 4 : développer un « checker » qui prend en entrée le fichier de sortie de vos programmes et vérifie la validité des contraintes de coupe et de la valeur des solutions trouvées.

Tâche 5 : Votre travail sera synthétisé dans un rapport qui contiendra votre analyse du problème, les schémas conceptuels de données, les algorithmes ainsi que leur complexité algorithmique (en temps, et en mémoire si besoin est), des tests et leurs résultats numériques (structures et valeurs des solutions, temps CPU, etc.).

Ce travail est à effectuer par **groupe de 3 à 4 étudiants**.

Date de rendus : vendredi 06/03 ou dimanche 08/03 ou vendredi 20/03 sur Teide.

Pour les rendus, chaque groupe met à disposition un fichier archive compressé de type GroupeX.zip (X est le numéro du groupe) que vous téléverserez sur Teide. Je créerai 6 groupes sur Teide.

Ce fichier archive doit contenir la structure de répertoires suivants:

GroupeX/Programs/ // répertoire contenant vos programmes sources

GroupeX/Report/ // répertoire contenant votre mini-rapport (10-15 pages max.)
reprenant et détaillant les points clés de votre présentation et de votre projet.

GroupeX/Data/ // répertoire contenant vos fichiers de données

GroupeX/Results/ // répertoire contenant vos fichiers de résultats expérimentaux

GroupeX/Articles/ // répertoire contenant les articles de la littérature si vous en avez
les .pdf, .docx, etc. Pensez à les référencer dans votre rapport

REMARQUE : Si vous devez changer des spécifications des fichiers ou faire d'autres hypothèses, c'est possible. Il faut simplement les préciser clairement dans votre rapport.