TP3(Rapport complémentaire)

Pour ce tp, on ne change pas les codes, juste ajoute des commentaires des fonctionnes.

Classe1 ***Job_shop( )****:* Pour résoudre la problème de NP, on prend ***jobshop_exhaust()*** comme exemple, au début il lit les fiches de tests selon le chemins, et il obtient la quantité de machines, quantités de taches, et la liste de tache(le temps d'arrive, la durée de ce tache). Et puis on mit on ordre tous les taches par le temps d'un tache fini de petit au grand. Maintenant, on utilise la fonctionne ***ajouter_possibilite( )*** pour faire la liste de toutes les possibiles des taches(00000,00001,00010…), par exemple, 00101, c'est a dire le tache 0,1,3 est dans la machine0, tache 2,4 est dans la machine 2. Pour chaque possbilité(un certificat), on divise les taches en deux listes de taches selon leur machine par la fonctionne ***Verifier()***, pour les taches dans la même machine, on utilise ***compare()*** pour tester si on peut mettre toutes les taches et retarder moins de D secondes.

```
public static String compare(ArrayList<tache> taches, int d) {
        String analyse = "oui";
        if (taches.size() > 0) {
            int y = taches.get(0).debut + taches.get(0).duree;
            for (int i = 1; i < taches.size(); i++) {
                if (y <= taches.get(i).debut) {
                    y = taches.get(i).debut + taches.get(i).duree;
                } else if ((y > taches.get(i).debut) && ((y - taches.get(i).debut))
<= d) {// taches.get(i+1).setDebut(taches.get(i).debut+taches.get(i).duree);
                    y = y + taches.get(i).duree;
                } else {
                    analyse = "non";
                }   }   }
```

```
        return analyse;
    }
```

Dans la algorithme de Vérification de JSP, on juste demande l'utilisateur qui propose un certificat lui-même(par exemple "01001") Dans la algorithme de aléatoire, la programmation va proposer un cortical lui-même, et retourne le résultat.

Classe2 ***JSP_ins( )****:* Cette classe est l'objet de jsp, un objet de jsp contient la numéro de machine, numéros de taches ,la liste de taches, et le temps de "delay".

Classe3 ***Main( )****:* Contient tous les entrée de toutes les fonctionnes,

Classe4 ***Sum_ins() :***Cette classe est l'objet de Sum, un objet de Sum contient la somme des chiffres, la quantité de chiffres, et la liste de chiffres.

Classe5 ***Partition_ins() :***Cette classe est l'objet de Partition et la fonctionne de réduction, un objet de partition contient la quantité de chiffres, et la liste de chiffres.
La fonctionne ***JSP_ins resoudre_jsp(Partition_ins p)*** est pour réduire le Partition en JSP. Et la fonctionne ***Partition_ins resoudre_partition(Sum_ins s)*** est pour réduire le Sum en Partition.
Et la fonctionne ***JSP_ins resoudre_jsp(Sum_ins s)*** est pour réduire le sum en JSP.

```
    public static JSP_ins resoudre_jsp(Sum_ins s) {
        Partition_ins p = resoudre_partition(s);
        JSP_ins jsp = resoudre_jsp(p);
        return jsp;
    }
```

```
    public static JSP_ins resoudre_jsp(Partition_ins p) {
        List<tache> chiffres = new ArrayList<tache>();
        for (int s : p.chiffres) {
            chiffres.add(new tache(0, s, 0));
        }
        int num_machine = 2;
        int delay_max = 0;
        JSP_ins jsp_new = new JSP_ins(num_machine, p.num, chiffres,
delay_max);
        return jsp_new;
    }
```

```
public static Partition_ins resoudre_partition(Sum_ins s) {
        Partition_ins partition_new = new Partition_ins(s.num_chiffre,
s.chiffres);
        return partition_new; }
```

Classe6 ***Partition() :*** Comme JSP, on teste toutes les possibilités, pour chaque certificat, par exemple, 01011,cest a dire le chiffre 1,2 sont au gauche, 0,3,4 sont au droite, si la sommes de la gauche et la droite sont pareille, retourne un 'oui' ,sinon retourne un "non"

Classe7 ***Sum() :*** Comparé avec Partition, dans les données de SUM, il ajoute un somme. On teste toutes les possibilités, pour chaque certificat, par exemple, 01011, c'est à dire le chiffre 1,2 sont utilisés pas, 0,3,4 sont ajouté dans notre addition, si la sommes de les chiffres est comme la somme qui est donné par les données, retourne un 'oui' ,sinon retourne un "non"