

Travail Pratique Algorithmique

1.1) Maximisation espace utilisé

i) Le pseudo-code est le suivant :

```
table = [s1, s2, ..., sn]
table.sort(descending)
weight := 0, take = []
for i in range(0, len(table)) do
    if weight + table[i] < D then
        weight = weight + table[i]
        take.append(table[i])
    end
end
return take
```

ii) Non il ne donne pas une solution optimale pour les fichiers, par exemple, si le tableau des espaces trié est [9, 6, 4, 2] et que D est de 10, cet algorithme ne prendra que le fichier de poids 9, alors qu'on aurait pû également prendre les fichiers de poids 6 et 4, ce qui est une meilleure solution.

1.2)

i) À une modification près, il faut trier la liste dans le sens inverse, et prendre les fichiers les plus petits au début jusqu'à ne plus pouvoir en prendre.

```
table = [s1, s2, ..., sn]
table.sort(ascending)
weight := 0, take = []
for i in range(0, len(table)) do
    if weight + table[i] < D then
        weight = weight + table[i]
        take.append(table[i])
    end
end
return take
```

ii) C'est un algorithme optimal, car on prends les fichiers les plus petits, il n'existe pas de meilleure solution, sinon elle contiendra des fichiers plus grands ou égaux, donc on en aurait moins ou autant.