

1. Tâches

Bonjour, dans cette deuxième partie de l'informatique et ses fondements, nous allons parler d'algorithmiques. Cette partie sera découpée en quatre séquences qui seront d'une part, une partie introductive sur les tâches, comment, à partir d'un problème compliqué, on peut le découper en petits morceaux. Ensuite, une partie sur les variables. Donc, qu'est-ce qu'une variable ? Qu'est-ce qu'on peut en faire ? Ensuite, une partie sur les instructions élémentaires, c'est-à-dire, un algorithme est composé de petits morceaux, quels sont exactement ces morceaux ? Et enfin, on parlera de culture algorithmique, c'est-à-dire on reviendra de façon plus générale sur ce qu'est un algorithme et qu'est-ce qui existe en algorithmique.

Nous allons donc commencer par les tâches. Alors, pour expliciter les tâches, je vais commencer par un petit algorithme, un petit problème simple. Donc, on a un tableau de notes donc, on a les notes d'un certain nombre d'élèves qui sont des valeurs à virgules. La question c'est, étant donné ce tableau, combien d'élèves ont la moyenne ? Sachant que le tableau peut être grand. Donc là, dans ce cas particulier, c'est assez facile parce qu'en fait, on va regarder les notes qui sont plus grandes que dix, là, elles sont colorées en rouge sur le transparent et on va les compter avec le doigt et donc là, on voit bien qu'il y a six notes qui sont au dessus de dix donc, il y a six élèves qui ont la moyenne. On a vu sur cet exemple qu'on pouvait le faire à la main. Maintenant, c'est plus intéressant de le faire avec un ordinateur donc, qu'est-ce que sait faire l'ordinateur ? Il sait faire des tas de choses : il sait faire des calculs, il sait éventuellement le mettre dans des variables, dans des endroits de la mémoire, il sait exécuter des tâches éventuellement, successivement, il sait faire des tests, il sait faire des boucles. Donc, on repassera sur tout cela dans d'autres séquences. Mais en fait, ce qu'il faut retenir, c'est qu'il sait faire des tâches simples et unitaires, c'est-à-dire, il sait faire un ordre successivement très simple. Donc, revenons à notre exemple : on a un tableau très, très grand, de millions de millions de cases, comment on va faire ? Là, on n'a pas envie de colorer en rouge les cases et de les compter à la main. En fait, si on avait qu'une seule case, ce serait facile parce qu'on la regarde. Si c'est plus que dix, ça veut dire que la réponse, c'est un et sinon, c'est que c'est zéro. Alors, imaginons qu'on sache faire la solution quand on a $n-1$ cases. À ce moment-là, sur les $n-1$ cases, il y a s élèves qui ont la moyenne, à ce moment-là, je n'ai plus qu'à regarder la n ième, si c'est plus grand que dix, c'est que j'ai $s+1$ élève qui ont la moyenne et si la note, elle est plus petite que dix, je n'ai que s élèves qui ont la moyenne. Donc ça, c'est un découpage du problème en tâches unitaires et simples. Et donc, on va en faire un algorithme qu'on va écrire dans un pseudo-code. Donc là, ce n'est pas encore de la programmation, c'est encore de l'algorithmique. Donc, un programme va avoir des entrées. L'entrée, c'est tout d'abord l'entier, qui est la taille du tableau que j'ai demandé à être strictement positive et ensuite, c'est le tableau lui-même qui est composé d'un certain nombre de cases de valeurs à virgules, peu importe. J'ai choisi que les cases sont $t[0]$ à $t[n-1]$, ça dépend du langage, mais là, c'est mon pseudo-code, j'ai choisi que ça va de 0 à $n-1$ et en sortie, j'ai l'entier s qui est le même que sur le transparent précédent qui dit le nombre d'élèves, nombre de valeurs qui sont plus grandes que dix. Ensuite, j'ai l'algorithme donc, j'ai ma valeur s que j'initialise à zéro. Et ensuite, j'ai une variable i qui va me servir à parcourir mon tableau et donc, je vais parcourir successivement chaque case de mon tableau. Et pour chacune des cases, je regarde la valeur : si elle est plus grande que dix, j'ajoute un à la variable s et si c'est plus petit que dix, surtout je ne fais rien. Et donc, une fois que j'ai fini de parcourir tout le tableau, j'ai le nombre de valeurs qui sont plus grandes que dix.