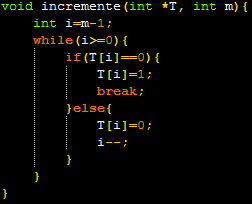
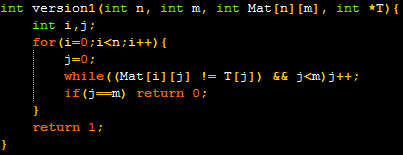
**Rapport N° 6**

***Le problème SAT***



La fonction incrémente, prend un tableau puis ajoute 1 de tel sorte qu’il reste un tableau de 0 et de 1 en binaire.

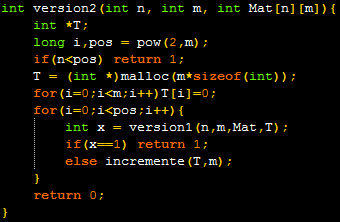
1. **Version 1 :**



On passe un tableau binaire et la matrice de clauses puis vérifier si le problème est satisfaisable avec ce dernier.

La complexité : n\*m = O (n²)

1. **Version 2 :**



On passe la matrice de clauses puis vérifier toutes possibilités de la table de vérité, si l’une d’eux retourne vrai alors on arrête et donc le problème est satisfaisable.

Pour la complexité :

* Si n < 2m: m + 2 = O(n) linéaire
* Si n >= 2m: = m(n\*(n+1)/2) = O(n3)

Pourquoi le problème est satisfaisable dans le cas n < 2m?

Dans la disjonction dans une clause, n’importe qu’elle clause soit fausse que dans un seul cas et vrai dans tous les autres cas.

Ex : (a et b) fausse pour (0,0) et vraie pour (0,1) (1,0) (1,1).

Nombre de clauses possibles pour un nombre m de littéraux est de 2m , et si le nombre de clauses est inférieur à ce dernier, alors les 2m combinaisons ne peuvent pas toutes être fausses si on prend la remarque d’avant.

Ex : On prend 2 littéraux a, b. Nombre de clauses possibles est 2²=4.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Clause | (0,0) | (0,1) | (1,0) | (1,1) |
| a et b | F | V | V | V |
| ¬a et b | V | V | F | V |
| a et ¬b | V | F | V | V |
| ¬a et ¬b | V | V | V | F |

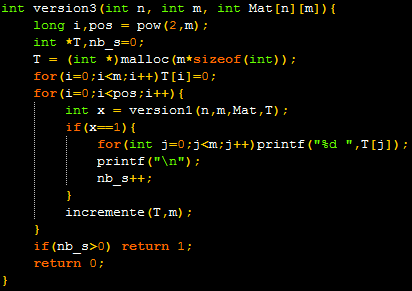
Si on supprime l’une des clauses, on obtient une table comme ceci :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Clause | (0,0) | (0,1) | (1,0) | (1,1) |
| a et b | F | V | V | V |
| ¬a et b | V | V | F | V |
| ¬a et ¬b | V | V | V | F |

On voit bien que la combinaison (0,1) ne va pas vérifier fausse car la clause qui permet de la rendre faux est manquante.

Dans ce cas, si n < 2m le problème est satisfaisable.

1. **Version 3 :**



On passe la matrice de clauses puis vérifier toutes possibilités de la table de vérité, et afficher toutes les solutions possibles.

Complexité : = m(2m \*(2m +1)/2) = O(n\*2n)