Projet d'Automne: Me too ?

Pseudocode de l'évaluation des contextes

Entrée: *nbSim*, *nbPers*, *worldSize people* (tableau contenant la position et l'état des personnes)
Sortie: affichage de la *densité*, du *taux_vaccination* et de la *mediane*

```
1: simulations[nbSim] (création du tableau des durées de contamination)
2: pour personnes allant de nbPers à 1:
3:         pour vaccination allant de 0 à personnes-1 :
4:                 si vaccination est différent de 0 :
5:                         people[vaccination][ETAT] ← V
6:                 pour sim allant de 0 à nbSim :
7:                         simulations[sim] ← simulation
8:                 pour i allant de 0 à personnes - 1 :
9:                         people[i][ETAT] ← N
10:                mediane ← mediane(simulations)
11:                densité ← personnes/world_size^2
12:                taux_vaccination ← vaccination/personnes
13:                afficher densité    taux_vaccination    mediane
```

L'évaluation des contextes se fait par trois boucles "for" imbriquées les unes dans les autres. Les boucles font varier ces paramètres : *vaccination* désigne l'indice de la personne à vacciner avant le début des simulations, *personnes* est passé en argument à **simulation** c'est l'indice de la première personne qui ne sera pas prise en compte dans la simulation et *sim* est l'indice de la simulation actuelle. La fonction **simulation** est elle-même découpée en fonctions plus petites: **but_check**, **new_but**, **blocage**, **move_find** et **move** s'occupent des déplacements et **test_incubation** ainsi que **test_contamination** s'occupent de la partie contamination.

La complexité de l'évaluation des contextes en fonction de *nbPers* est:

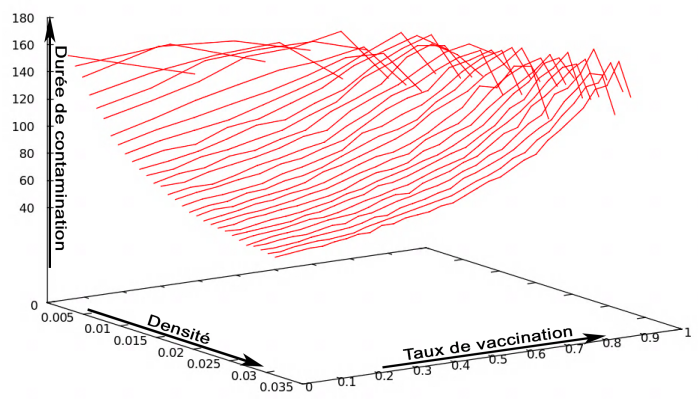
$$\mathbf{O}(nbPers^2)$$

car il y a deux boucles l'une dans l'autre qui s'éxécutent *nbPers* fois. La complexité du programme entier est:

$$\mathbf{O}(nbPers^4)$$

car à l'intérieur des boucles de l'évaluation des contextes, dans la fonction **simulation**, il y a encore la boucle de la mise à jour asynchrone qui itère sur *nbPers*, à l'intérieur de laquelle se trouve la boucle de **move_possible** et la boucles de **test_contamination** qui itèrent sur *nbPers*. Puisque ces boucles sont en parallèle on ne les compte qu'une fois ce qui donne un total de quatre boucles "for" sur *nbPers* les unes dans les autres.

Résultat du test12

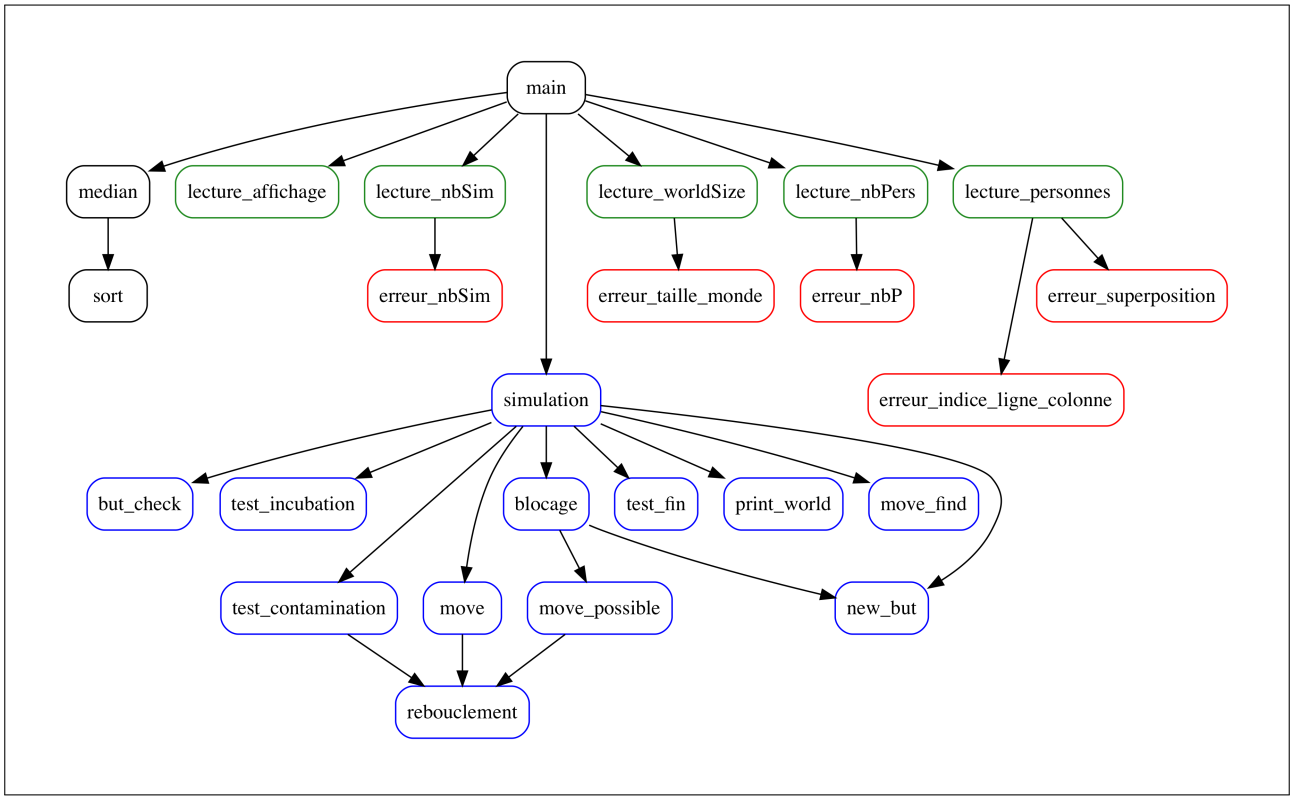


Ce graphe est obtenu à partir du fichier *test12.txt* (nbSim = 1000 pour un graphe plus lisse), le graphe montre la durée de contamination totale en fonction de la densité et du taux de vaccination. Sur le graphe on peut voir que la durée de contamination totale augmente lorsque la densité diminue et lorsque le taux de vaccination augmente. Quand la densité diminue le temps de contamination diminue car dans un monde moins dense les personnes ont moins de chances de se croiser et donc la probabilité qu'une personne contamine une autre est plus faible. Lorsque le taux de vaccination augmente le temps de contamination diminue également car les personnes contaminées ont moins de chances de croiser une personne susceptible d'être contaminée, il y a donc également une probabilité de contamination plus faible.

Dans la réalité il serait impensable de réduire la densité de population en supprimant des personnes, c'est donc la méthode de la vaccination qui est utilisée pour empêcher une maladie de se répandre. De plus on peut voir sur le graphe que l'augmentation du temps de contamination en fonction du taux de vaccination n'est pas linéaire, c'est que lorsque presque toutes les personnes sont vaccinées que le graphe grimpe beaucoup. Cela montre l'importance de vacciner le plus de personnes possible afin d'entraver au maximum la propagation des maladies.

Sur le graphe, lorsque le taux de vaccination est maximal, le temps de contamination total diminue. Cela semble incohérent à première vue mais en analysant la situation on voit que dans la situation ou une seule personne est non-vaccinée la condition de contamination totale est juste de contaminer cette personne. Pour terminer la simulation il suffit que la personne non-vaccinée croise la personne contaminée. La chute sur le graphe ne signifie donc pas que la "maladie" se propage plus vite mais que dans cette situation particulière les conditions de fin sont plus facilement atteignables.

Graphe des appels de fonctions



Voici le graphe des appels de fonctions, en bleu les fonctions appartenant a la simulation, en rouge les fonctions d'erreur et en vert les fonction de lecture des données de l'utilisateur.