

Dans l'archive contenant ce .pdf est aussi contenu un fichier .java contenant le code permettant de répondre au problème posé.

En voici les commentaires :

La variable `f`, déclarée ligne 24 et initialisée ligne 31, contient la liste des fichiers et dossiers présents dans le dossier où s'exécute le programme.

Parmi cette liste, il ne faut garder que le fichier contenant une liste de patrons aléatoires. Ce fichier doit s'appeler `N_random_nets.txt`, `N` étant mis pour le nombre de patrons contenus dans ce fichier. Il y en a deux exemplaires dans l'archive.

Le seul fichier qui nous intéresse se termine donc par « `random_nets.txt` ».

Une fois ce fichier identifié (à la ligne 34 donc), il s'agit d'en extraire les `N` patrons qu'il contient. C'est l'objet des lignes 36 à 47. La méthode `FileReader(String filename)` pouvant générer une exception (fichier non trouvé), il faut l'englober dans un bloc `try{...} catch{...}` afin de parer à cette éventualité.

Ensuite, tout se joue dans les lignes 40 et 41 où, dans l'ordre :

- La commande `RealLine()` lit une ligne de `in` ;
- Cette ligne est stockée dans la variable `line` ;
- Cette variable est comparée à la chaîne de caractères vide et l'intérieur de la boucle est exécuté si `line` n'est pas une chaîne de caractères vide ;
- Auquel cas, on ajoute au tableau `nets` la variable `line`.
- Sinon, la boucle s'arrête car on a atteint la fin du document.

La variable `nets` est donc maintenant un tableau (unidimensionnel) contenant les patrons.

On note `N` sa taille (ligne 51).

Ensuite (ligne 55 à 61), pour chaque élément `net` de `nets`, on convertit cette chaîne de caractères en tableau, en découpant selon le caractère ",". La variable `path` est alors un tableau contenant 62 chaînes de caractères (chacune correspondant au nombre qu'elle représente).

Pour chaque élément `s` de `path` (lignes 58 à 60), on commence par convertir la chaîne de caractères `s` en le nombre entier qu'elle représente et on ajoute 1 à la `Face` en question.

À la fin de cette boucle, `Face` est un tableau de 32 cases tel que : `F[k]` est le nombre d'apparition de la face `k` dans la suite des `N` patrons aléatoires.

Il ne reste plus qu'à afficher les fréquences correspondantes en divisant par `62N`. À noter que le `62.0*N` permet de faire une division décimale, c'est-à-dire d'obtenir un résultat décimal. Si l'on avait écrit `62*N`, les nombres affichés auraient été des parties entières donc ici uniquement des 0.

Il ne reste plus qu'à faire un magnifique diagramme en bâtons que l'on peut admirer dans le fichier .xlsx joint.

CONCLUSION

Les faces 1 à 5 (et d'autres) possèdent les mêmes fréquences d'apparition. Rien de bien étonnant puisque ces faces sont au même endroit dans le polyèdre, à une rotation de centre O et d'axe (Oz) près.

Les faces 0 et 31 sont seules à avoir des fréquences distinctes et là non plus, rien de bien étonnant de par leur position dans l'icosaèdre tronqué.

Quant à calculer les probabilités correspondant à ces fréquences... ceci est un autre problème...