Partie 2 :

Q9)

Combien de combinaisons de couleurs (secrètes) c'est un N-arrangement sans répétition de N couleur parmi K-couleur .

$$\boxed{A_K^N = \frac{K\,!}{(K-N)\,!} \quad si\ N<=K}$$

Q10)

si le joueur 1 est rependu (b, 0) alors on a b couleurs bien placer et le reste sont ni bien placer ni bien choisies , on va sélectionner les b couleurs possible qui sont bien placer et pour chaque sélection on a tout les arrangement possible des couleurs qui n'appartient pas a les N-couleurs choisies par le joueur 1.
pour sélectionner les b couleurs possible qui sont bien placer c'est une combinaisons de b couleurs parmi les N couleur choisies par joueur 1.
pour les arrangement possible avec chaque sélection on arrange N-b (le reste) parmi K-N (les couleurs qui n'appartient pas a les couleurs choisies par joueur 1).

E :combien de possibilités de combinaisons secrètes peuvent-elles être envisagées par le joueur 2

$$\boxed{E = C_N^b \; A_{K-N}^{N-b}}$$

Q11)

-on prendre b couleur parmi N couleurs choisi par le joueur 1 et pour chaque sectionnement possible on a une seul position (ordre) possible pour chaque couleurs alors c'est une combinaisons de b parmi N .

alors on a : C_N^b

- pour chaque combinaisons on prendre m couleurs parmi N-b (les couleurs qui sont choisi par le joueur 1 et qui ne sont pas bien placer) .

alors on a : C_{N-b}^m

- et pour chaque sélection possible on peut arranger les m couleurs parmi N-b couleurs (les couleurs qui sont choisi par le joueur 1 et qui ne sont pas bien placer).
mais on calcule seulement les arrangement tell que les m couleurs sont tous mal placer .

on note **F**   le nombre des arrangement possible tell que les m couleurs sont tous mal placer .

alors on a un nombre de cas moins que les tt arrangement possible de m couleurs

$$F \leq A^{m}_{N-b}$$

-et pour chaque sélection possible et arrangement des couleurs choisies et mal placer on peut arranger le reste qui sont mal choisies N – b – m (le reste qui ni bien choisies ni bien placer) parmi K-N (tout les couleurs – les couleurs choisies par le jouer 1) .

alors on a : A^{N-b-m}_{K-N}

E :le nombre de combinaisons secrètes compatibles avec une réponse **(b, m)**

$$\mathbf{E} = C^{b}_{N} \; C^{m}_{N-b} \; F \; A^{N-b-m}_{K-N}$$

$$\boxed{\mathbf{E} \leq C^{b}_{N} \; C^{m}_{N-b} \; A^{m}_{N-b} \; A^{N-b-m}_{K-N}}$$

Q12)

n'y a-t-il pas forcément égalité a cause de l’inégalité des arrangement tell-que les couleurs sont mal placer et tout les arrangement possible .

Exemple :
K=3, N=2
couleurs choisies par joueur 1 [1 , 2]
proposition de joueur 2 [2 , 1]
repense de joueur 1 (0,2)
combinaisons secrètes compatibles avec une réponse (0,2) on a un seul cas c’ est [2,1]
on calcule la formule sur le même exemple :

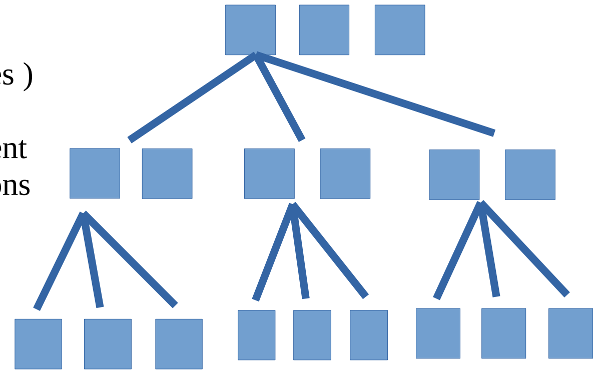
$$C^{0}_{2} \; C^{2}_{2-0} \; A^{2}_{2-0} \; A^{2-0-2}_{3-2} = A^{2}_{2} = \mathbf{2}$$

les 2 cas sont [2,1] et [1,2] le premier cas a comme réponse (0,2) mais pas le deuxième .

Programmation dynamique :

Q 13)

l'idée c'est pour chaque élément on a 3 cas possible (bien placer , bien choisies et mal placer , mal choisies) et pour chaque un des cas ce élément a un nombre de valeur possible , alors on peut considérer les n-1 élément qui reste comme une calcule de nombre de combinaisons secrètes pour la taille n-1 et k-1 ,et la valeur de b et m dépend a le cas de l'élément ,et pour le cas de base ou n vaut 0 on a 2 cas possible si b=0 et m=0 alors alors touts les choix précédant respect le nombre de b et m dans les combinaisons .



les conditions initiales :

$$c(k,0,b,m)=\begin{cases} 1 \text{ si } b=0 \text{ et } m=0 \\ 0 \text{ si non} \end{cases}$$

récurrence :

c(k,n,b,m)=
  c(k-1, n-1, b-1, m) // bien placer a un seul valeur possible
+ *c*(k-1, n-1, b, m-1) .(n-b) //bien choisies mal placer ce élément a n-b valeur possible
+ *c*(k-1, n-1, b, m)*(k-n) //mal choisies ce élément a k-n valeur possible

Q 14)

pour le calcule de nombre de combinaisons secrètes on a besoin de sauvegarder que les valeur de b et m de l'étape précédente .

```
pour b de 0 à N faire
     pour m de 0 à N faire
          c[b,m] = 0

c[0,0]=1
k=N-K
pour n de 1 à N faire
   k=k+1
   pour b de 0 à N faire
     pour m de 0 à N faire
        bien_placer=0
        si(b-1>=0)
            bien_placer=c[b-1,m]

        mal_placer=0
        si(m-1>=0)
            mal_placer=c[b,m-1]

        c[b,m]=bien_placer+ mal_placer *( n – b )+c[b,m]*(k-n)
résulta = c[B,M]
```

Q 15)

la valeur obtenue pour N = 4 , K = 6 , b = 1 et m = 2 est 72

Q16)

la complexité au pire cas de cet algorithme est $O(N^3)$

Q17) On ne peut pas utiliser le raisonnement précédent dans le cas d'un historique non vide, parce que contrairement au raisonnement précedent, l'espace de recherche est influencé par les proposition déjà dans l'historique ainsi qu'on doit énumerer la proposition pour qu'on puisse calculer le score.

Q18) Algorithme de calcul de score

```
CalculerScore( Proposition p, Historique H){

      HashMap<String, ArrayList<String>> list = new HashMap<>();
      Pour(chaque combinaison secrete possible s){
         Si (s n'est pas la meme que p) alors
           evaluer s  par rapport a p ; c-a-d calculer (b,m)
           ajouter(s,b,m) a la list ;
      }
         max = 0 ;
      Pour(pour chaque element E de la list){
         nbr = CombinaisonsSecretCompartible(E, H);
         Si(nbr > max){
           max = nbr ;
           garder E.b E.m comme meilleur proposition ;
         }
      }
      retourner (p, p.b, p.m) ;
   }
```

Q19) La complexité au pire cas de l'algorithme est : $O((A_k^n)^2)$

Q20)

```
Algorithme glouton()
Choisir une proposition p aleatiore au debut ;

Tantque( Evaluer (p) != (N,0)) {
        ajouter p a l'historique H
        meilleur ;
        Pour ( chaque combinaison secrete possible comb ) {
                score ← CalculerScore(comb, H)
                Si(score > meilleur.score) Alors
                        meilleur  = comb ;
                        meilleur.score = score ;
        }
        p = meilleur ;
}

SOUTION TROUVÉE
```

La valeur pour K=6 et N=4 , et une solution s=1234 en commaçant par 5426 est **56 coups**.