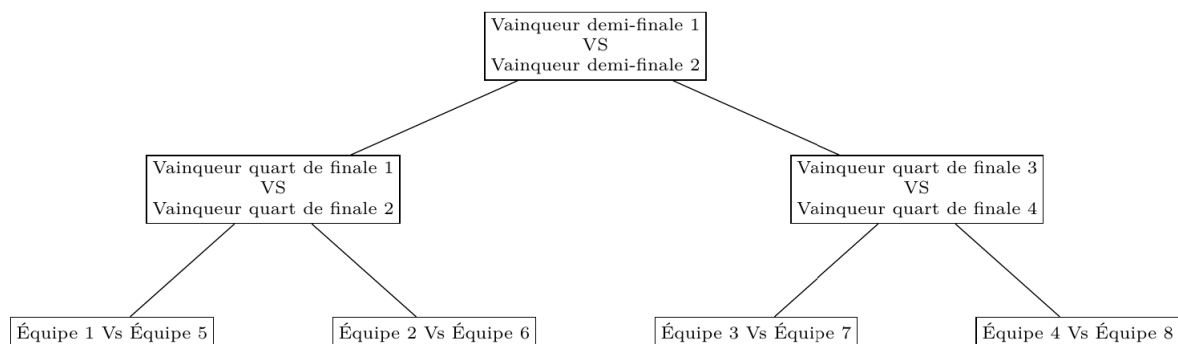
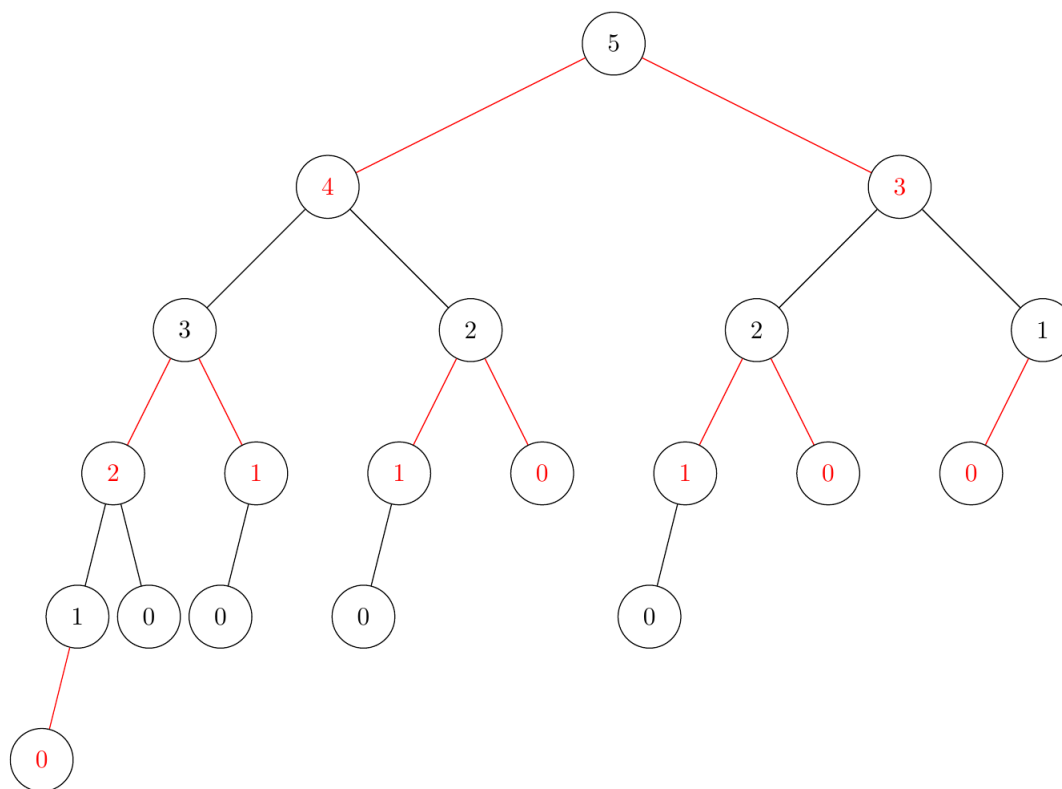


ARBRES BINAIRES - CORRIGÉ

Exercice 0

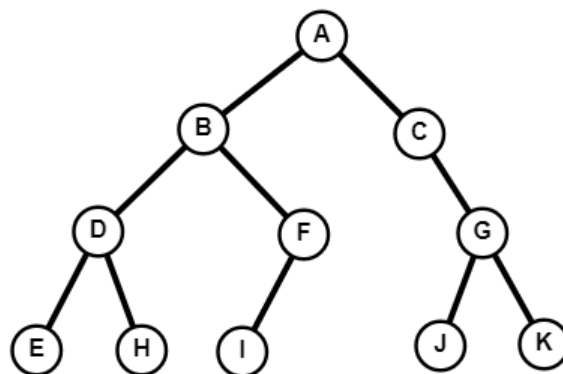


Exercice 1



1. Voir figure ci-dessus.
2. Chaque noeud possède au plus deux sous-arbres.
3. Chaque joueur a quatre possibilités de gagner ; elles correspondent aux noeuds de valeur 0, représentant le nombre d'allumettes restantes après avoir joué.
4. Si le 1er joueur prend deux allumettes (sous-arbre droit de la racine), alors quelque soit le nombre d'allumettes prises par le second joueur au second coup, il pourra atteindre le noeud de valeur zéro au coup suivant (sur le niveau 3 de l'arbre).

Exercice 2 On considère l'arbre binaire suivant :



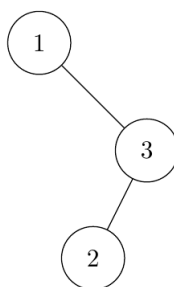
1. Il a 11 nœuds, soit $n = 11$.
2. Nœuds internes : {A, B, C, D, F, G}. Feuilles : {E, H, I, J, K} .
3. $h = 3$ (la racine est au niveau 0).
4. Fils du nœud B : {D, F}.
5. Sous-arbre gauche du nœud B : l'arbre de racine D.
6. Sous-arbre gauche du nœud B : $\emptyset$ (arbre vide)

Exercice 3

```

1 def affiche(a):
2     if a is None:
3         return
4     else:
5         print('(', end='')
6         affiche(a.gauche)
7         print(a.valeur, end='')
8         affiche(a.droit)
9         print(')', end='')
  
```

Exercice 4



Exercice 5

```

1 def __eq__(self, a):
2     return a is not None and self.gauche == a.gauche and self.droit == a.droit and
   self.valeur == a.valeur
  
```

Exercice 6

```

1 def parfait(h):
2     if h == -1:
3         return None
4     else:
5         return Noeud(parfait(h-1), h, parfait(h-1))
  
```