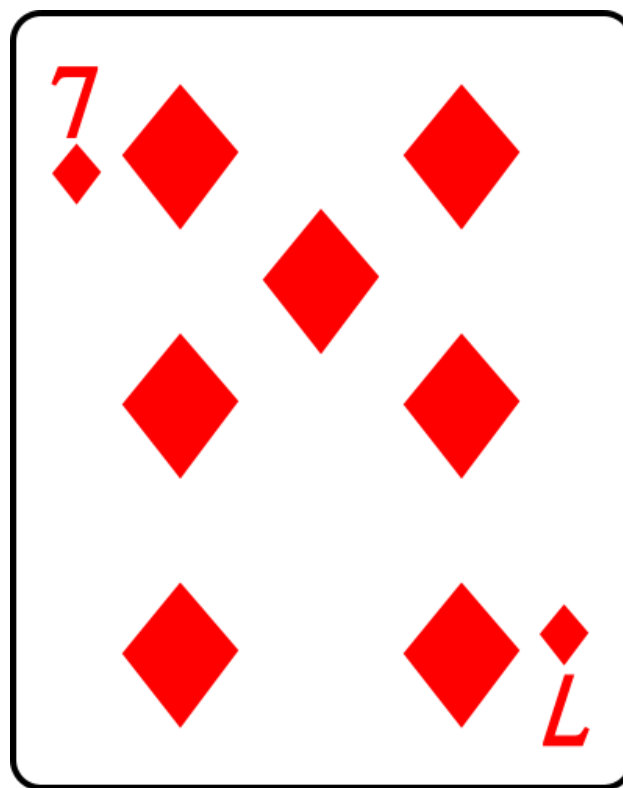


# Introduction à l'informatique CM6

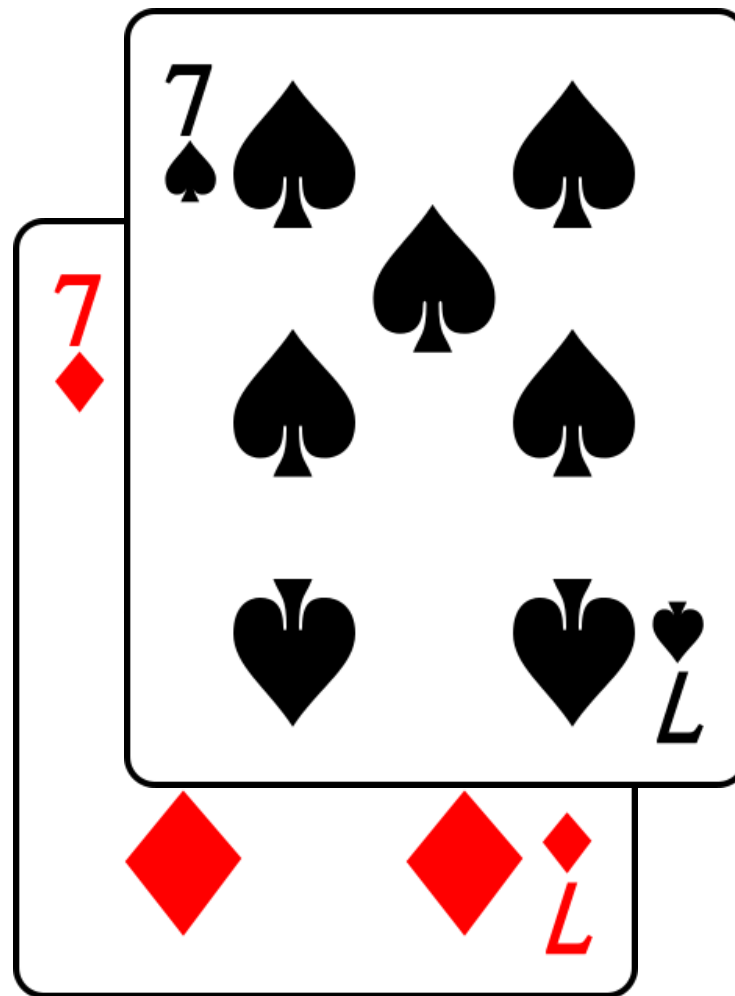
Antonio E. Porreca

<https://aeporreca.org/teaching>

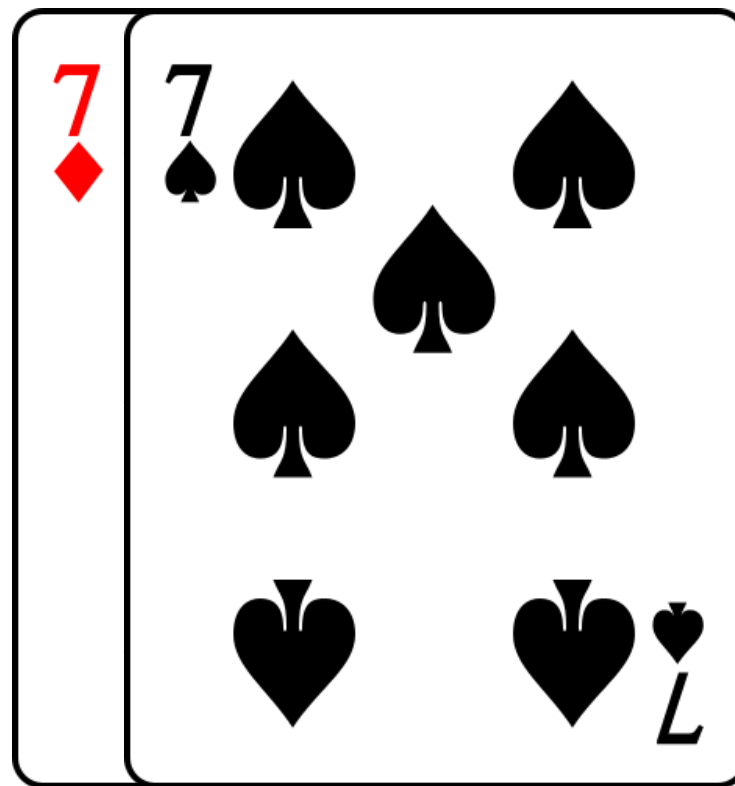
# Tri par insertion



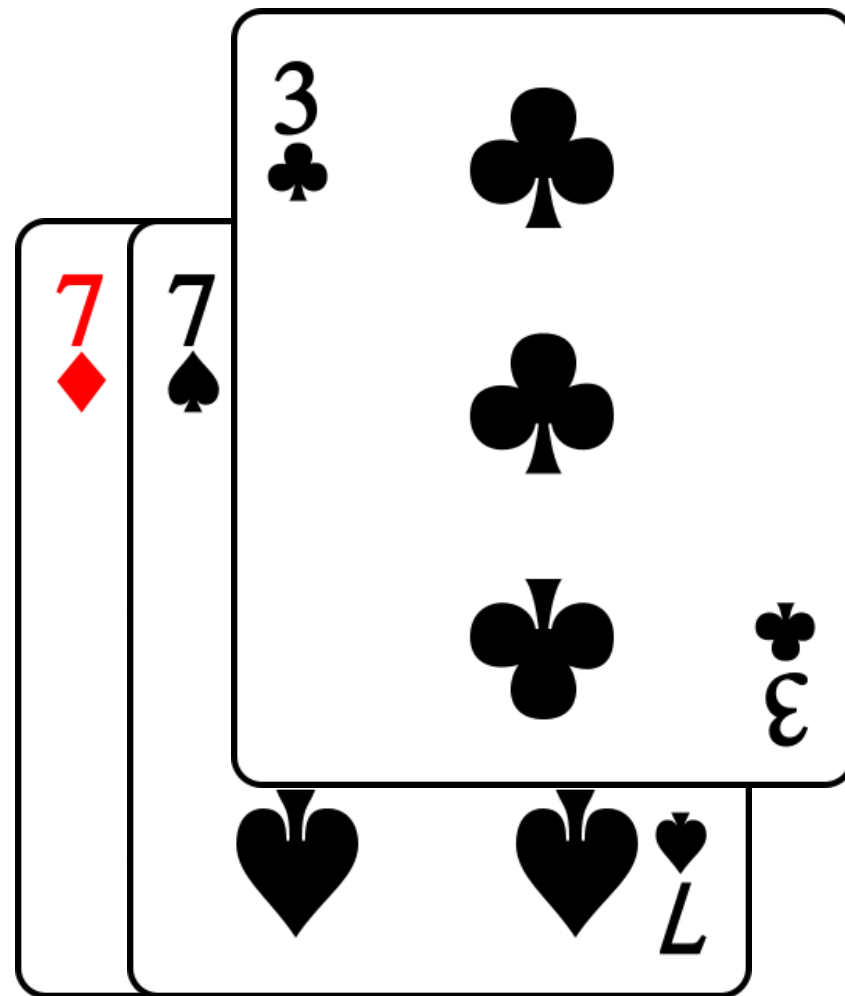
# Tri par insertion



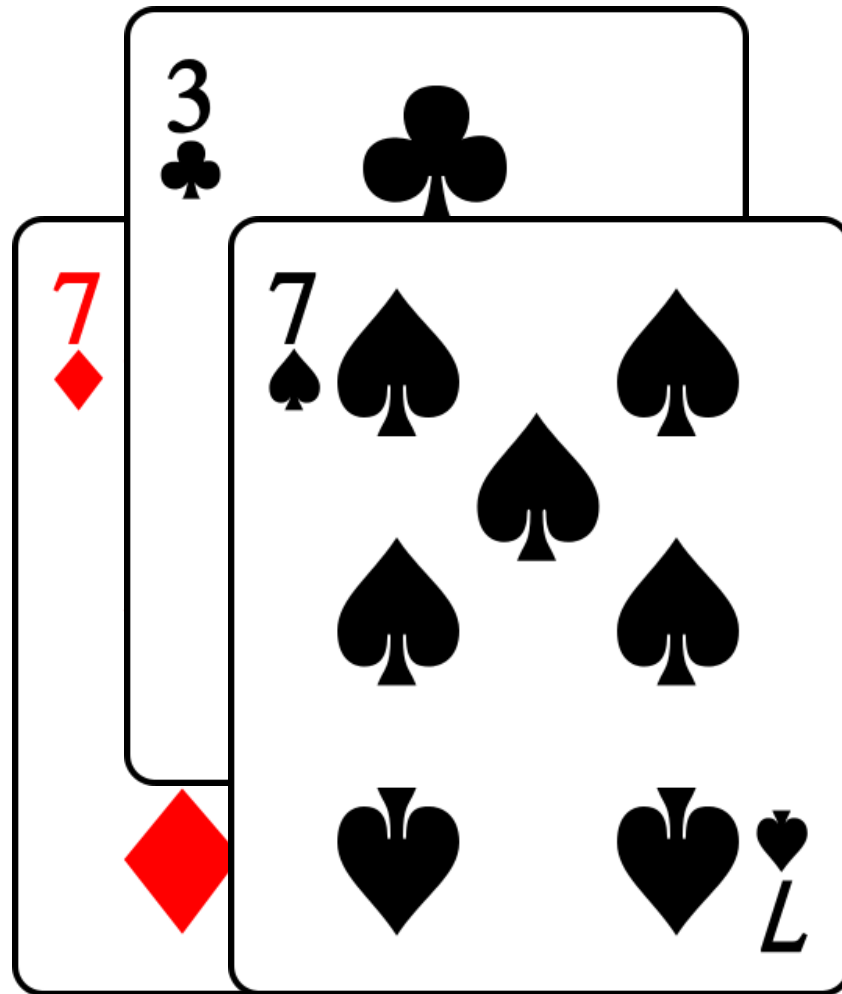
# Tri par insertion



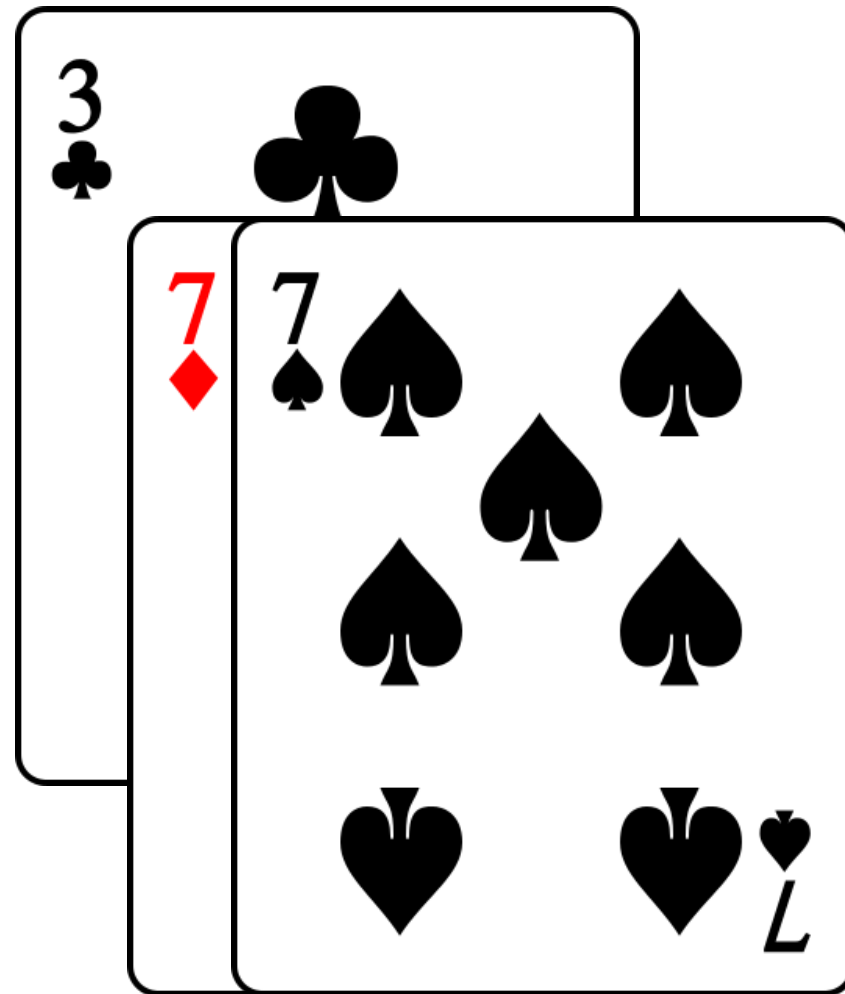
# Tri par insertion



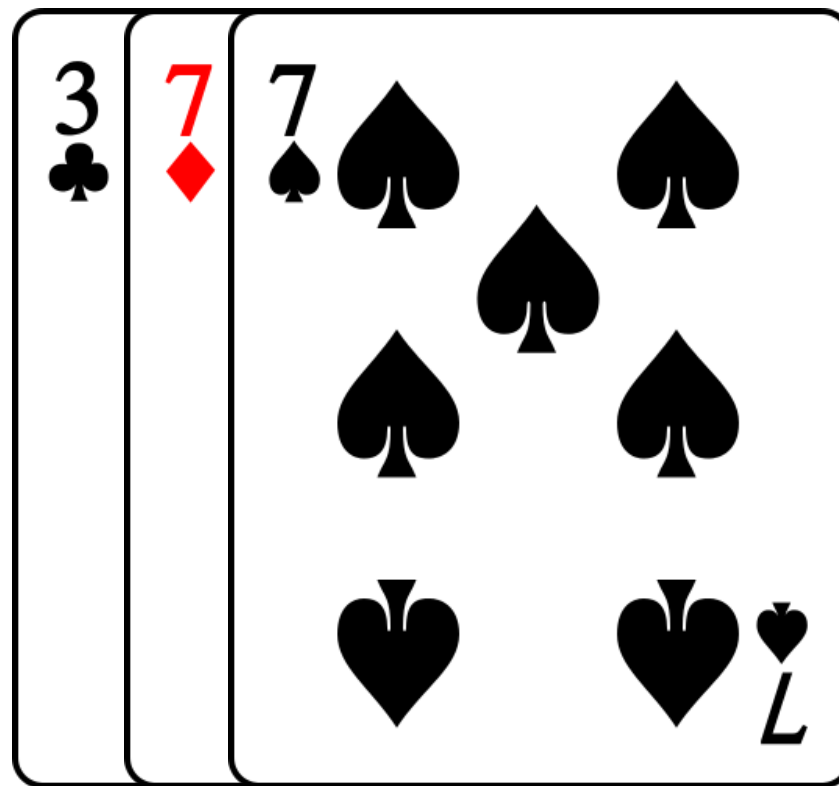
# Tri par insertion



# Tri par insertion

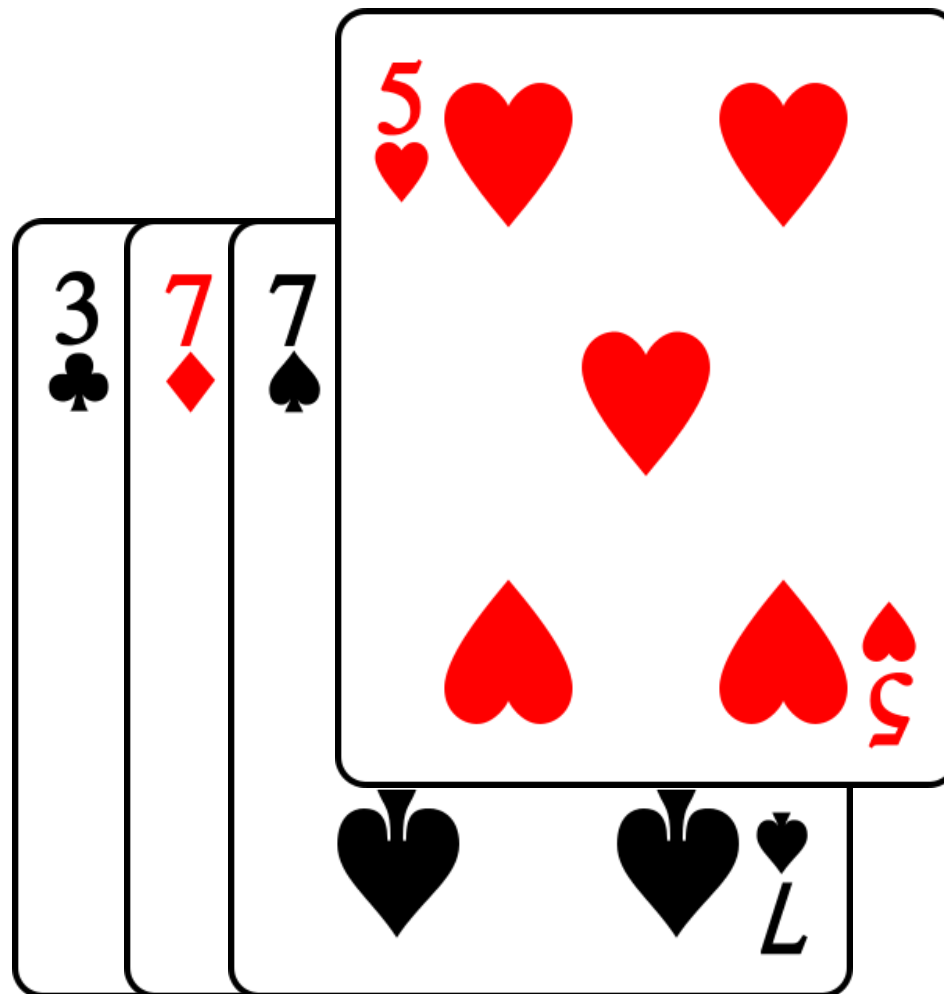


# Tri par insertion

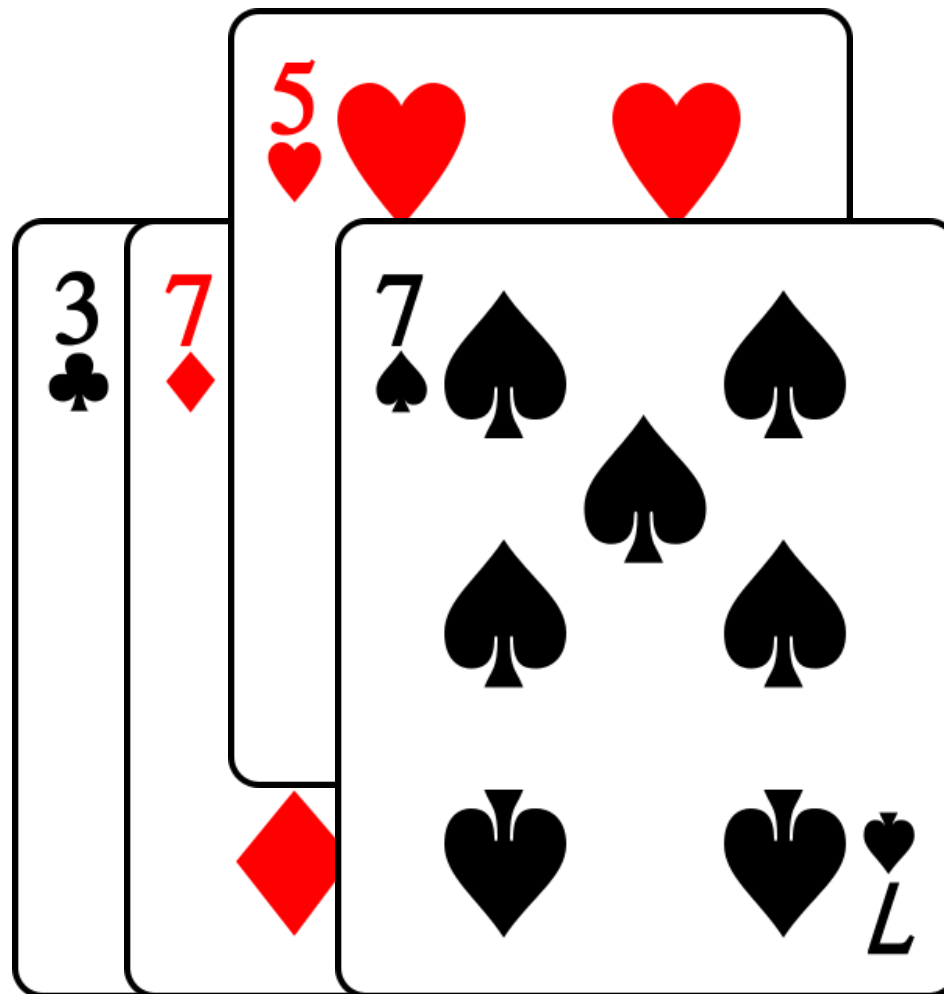




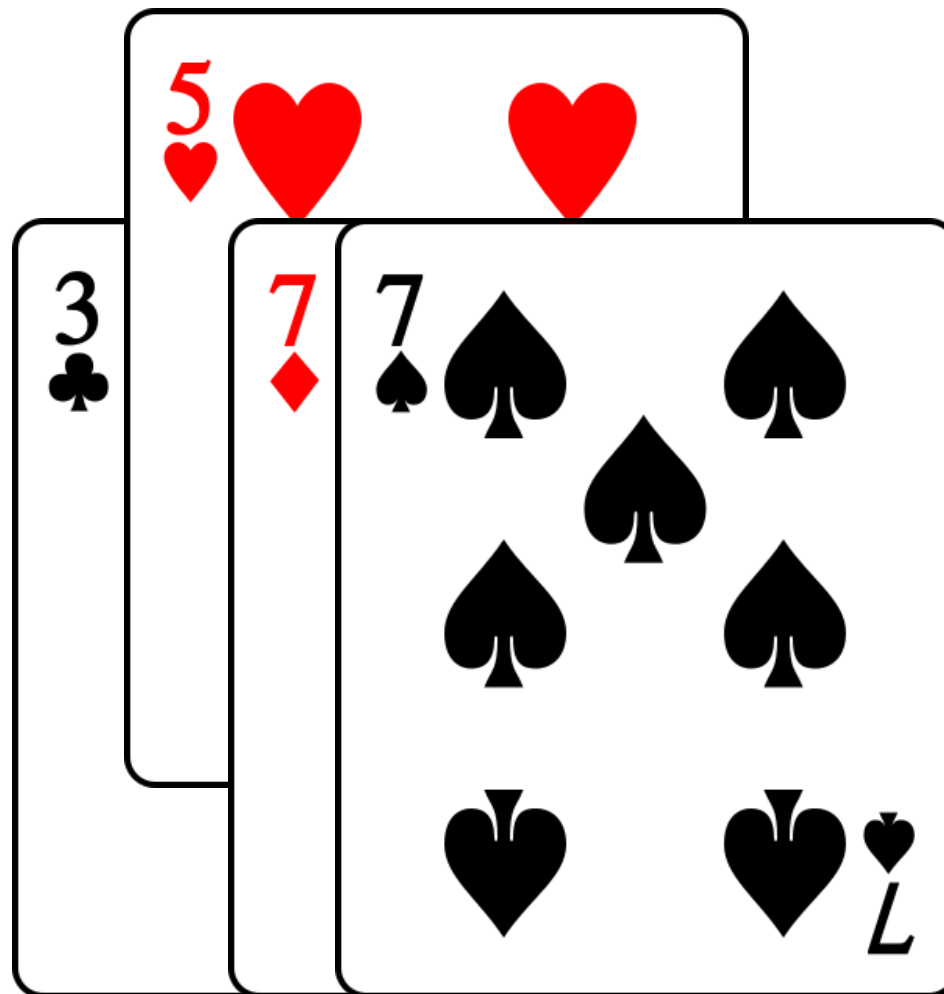
# Tri par insertion



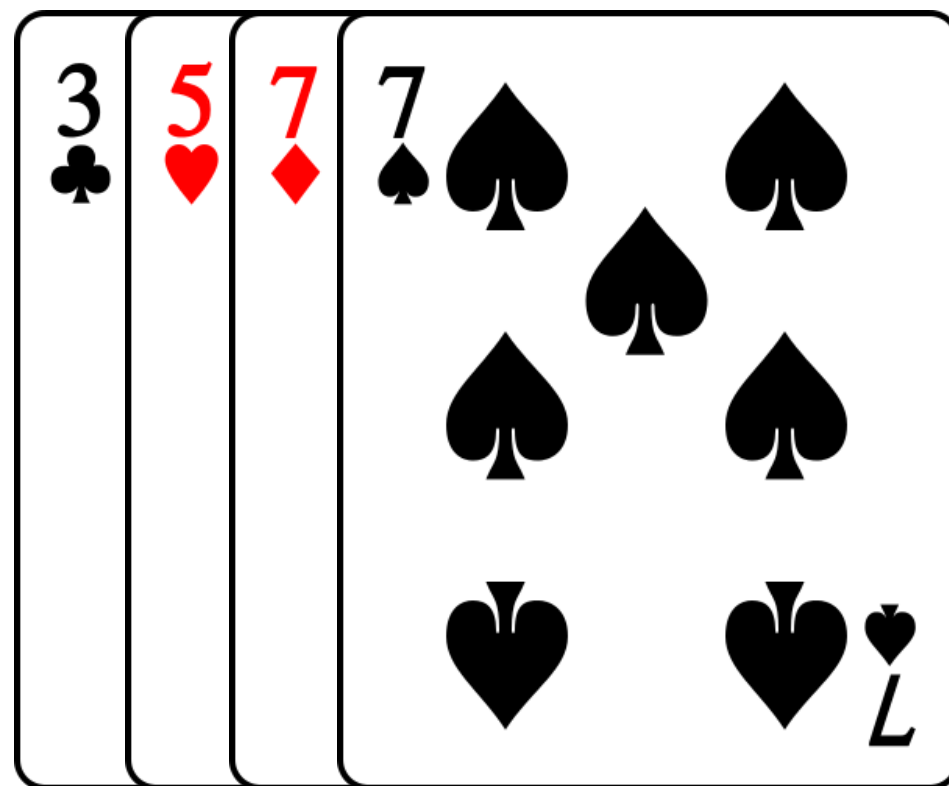
# Tri par insertion



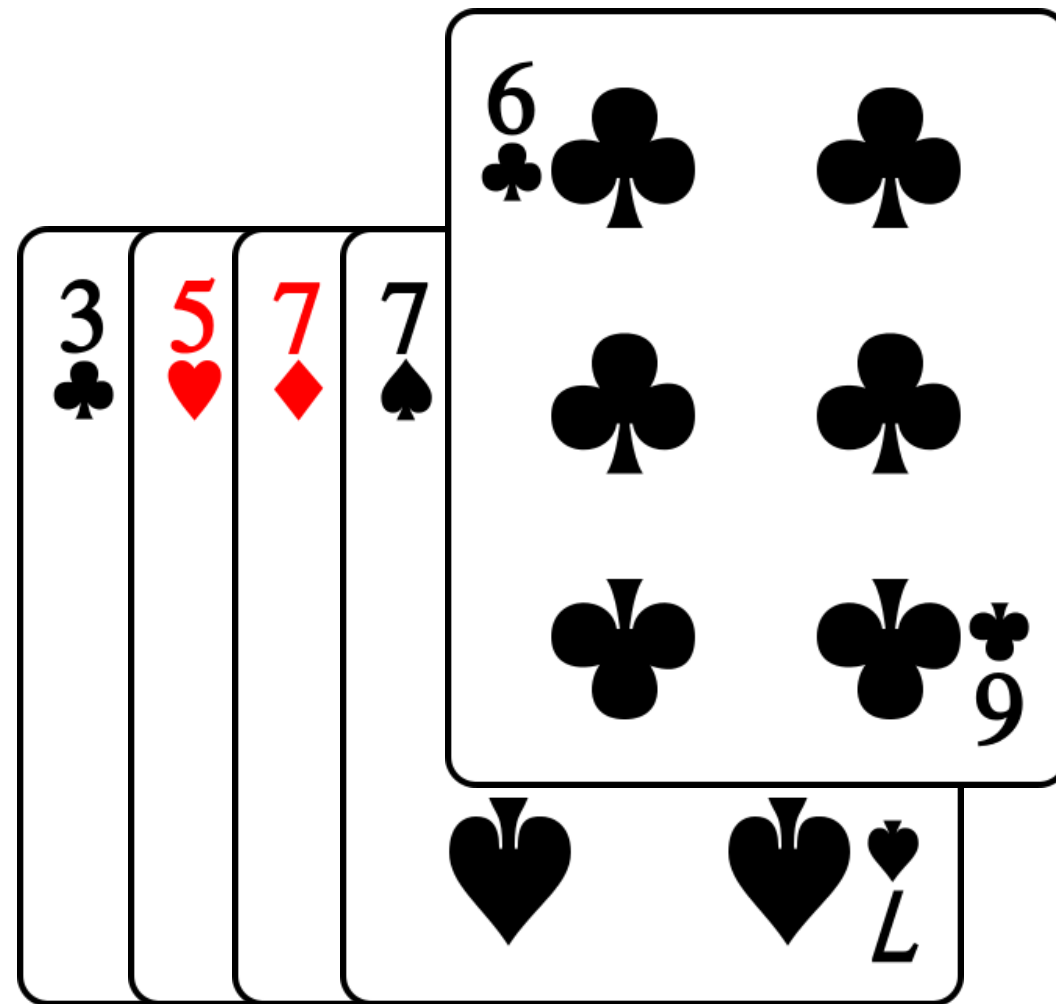
# Tri par insertion



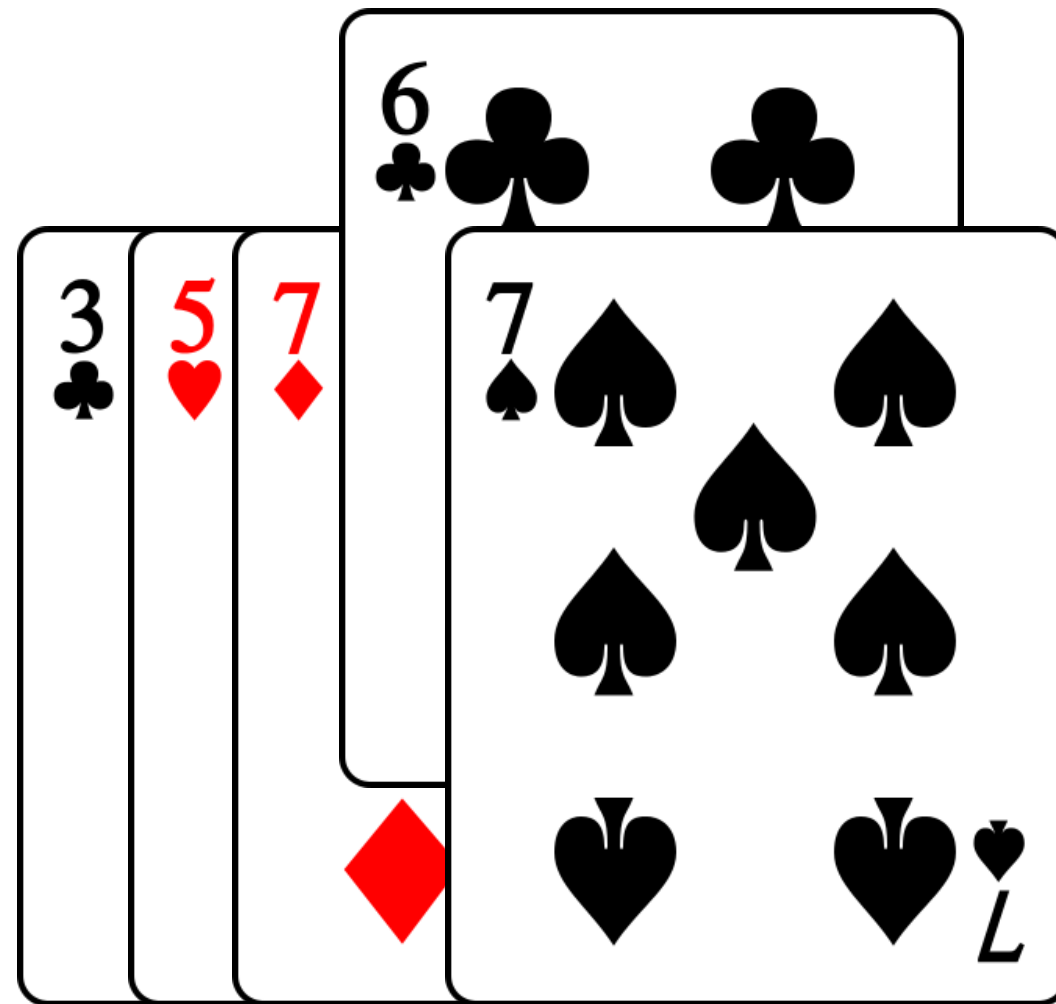
# Tri par insertion



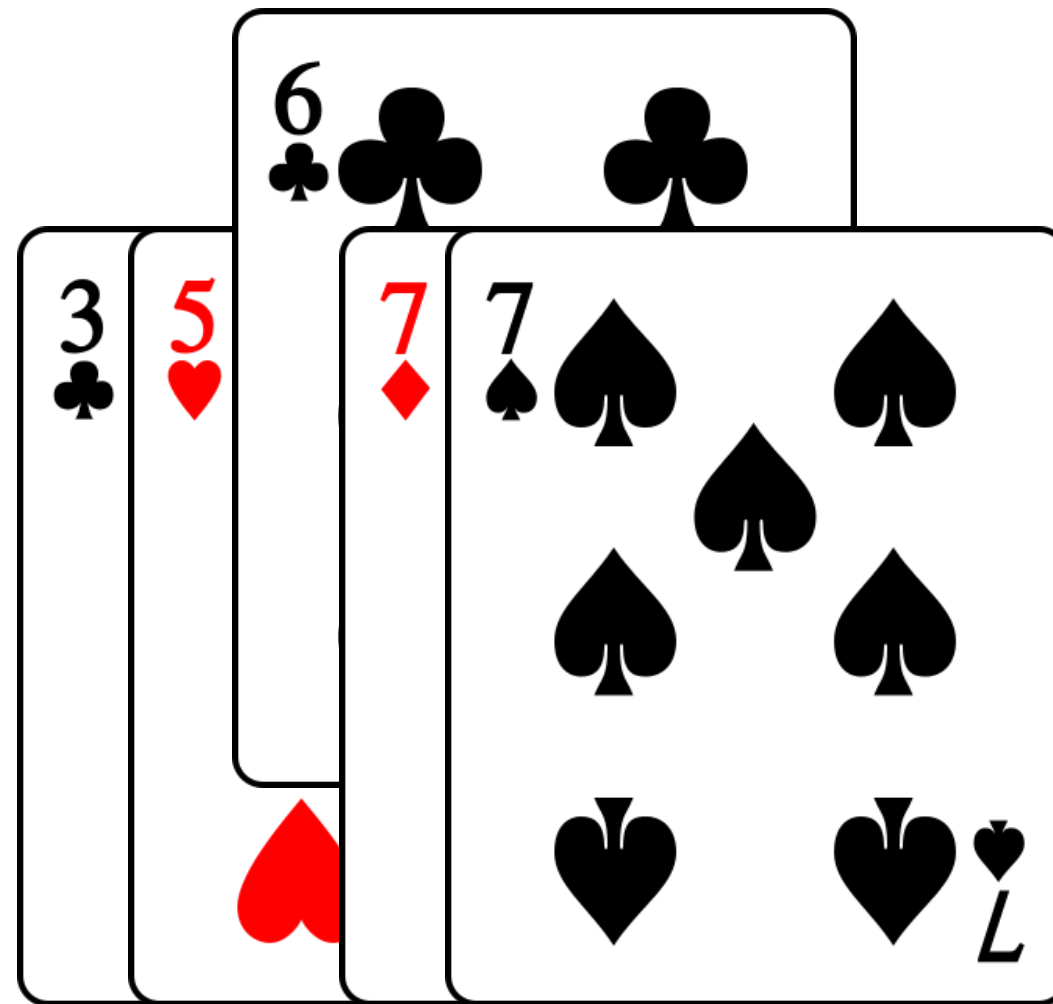
# Tri par insertion



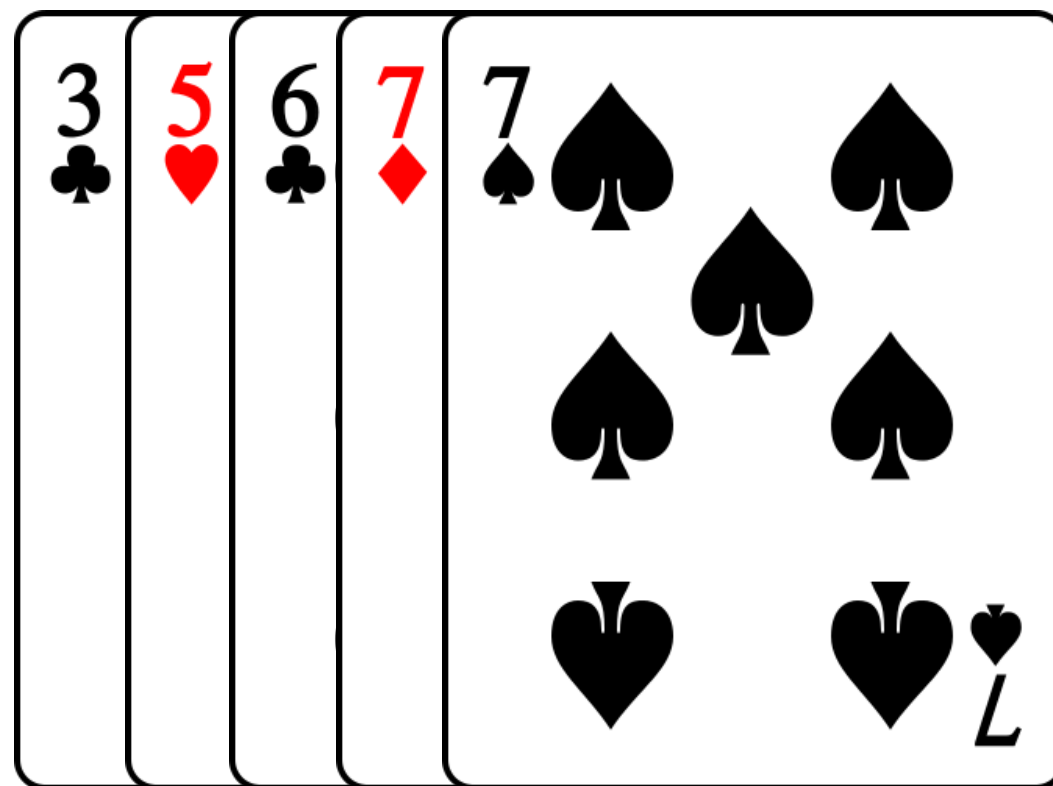
# Tri par insertion



# Tri par insertion



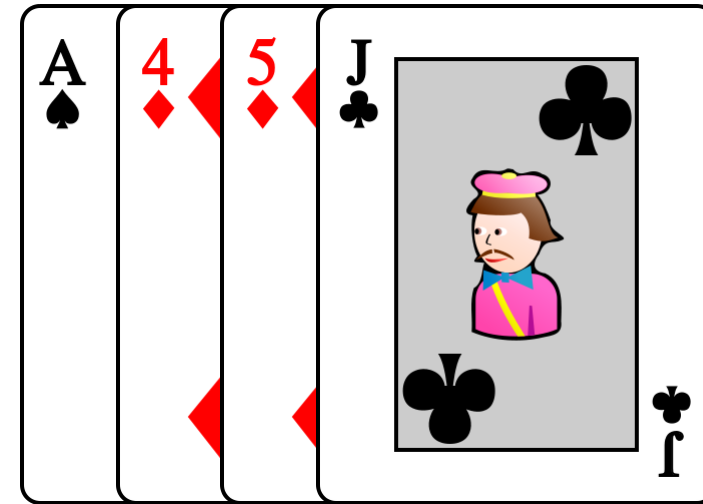
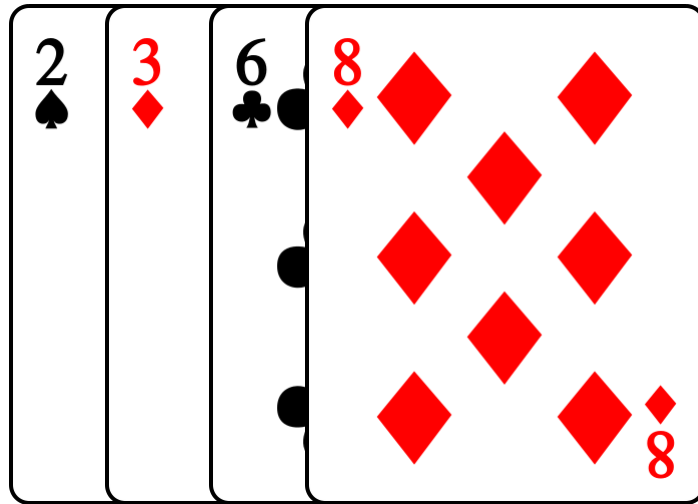
# Tri par insertion



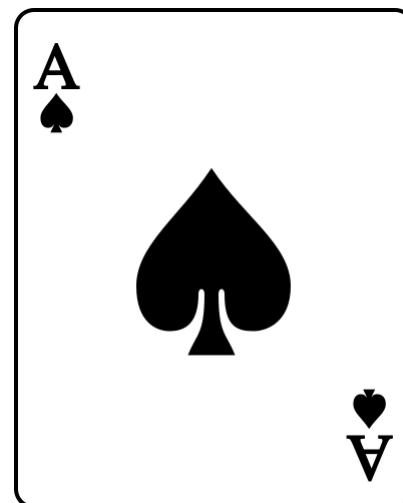
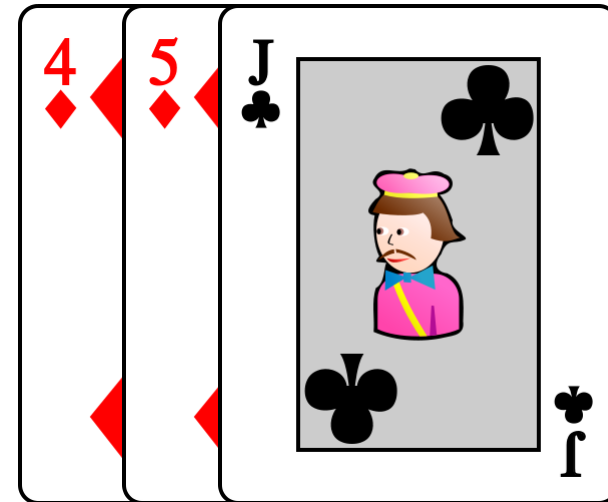
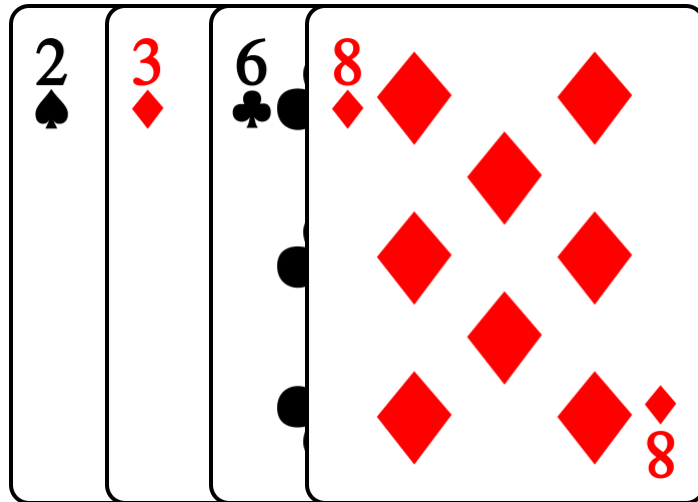


**Est-ce qu'on peut  
faire mieux que ça ?**

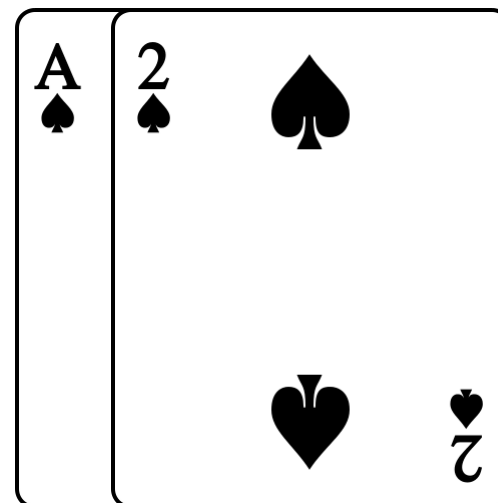
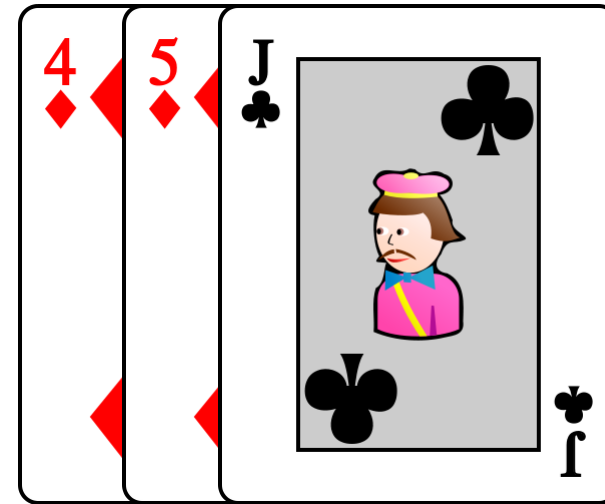
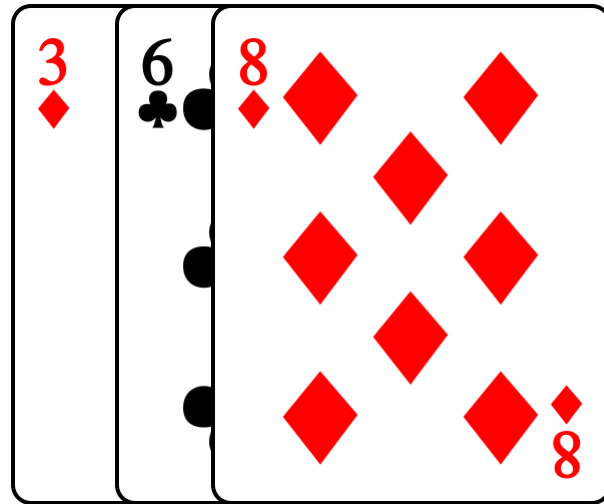
# Fusion de tableaux triés



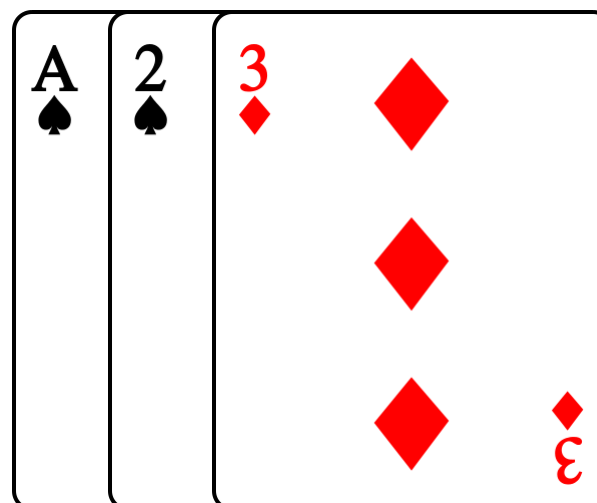
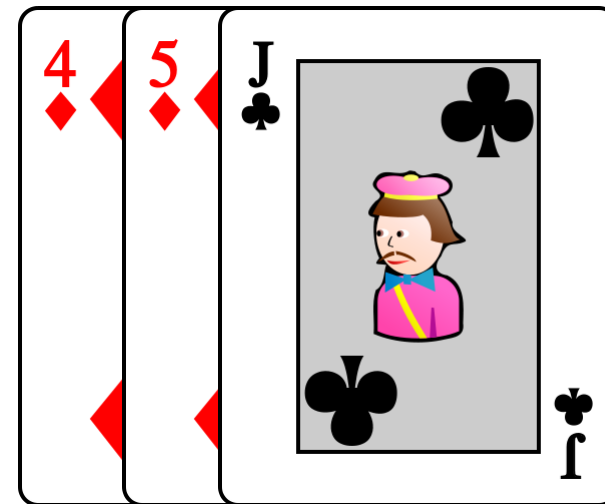
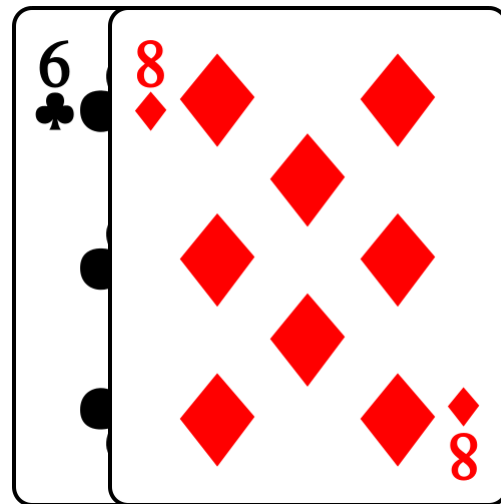
# Fusion de tableaux triés



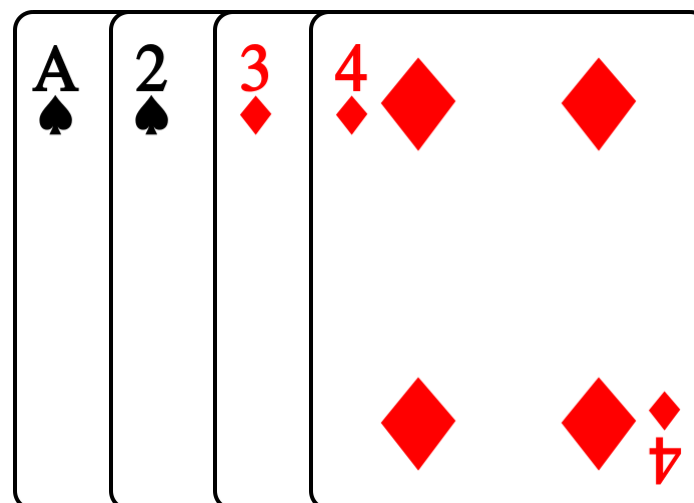
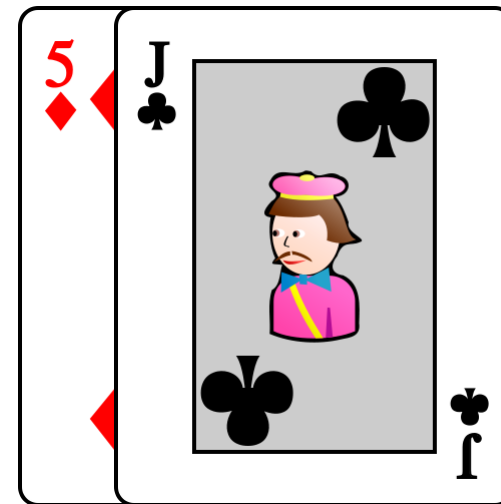
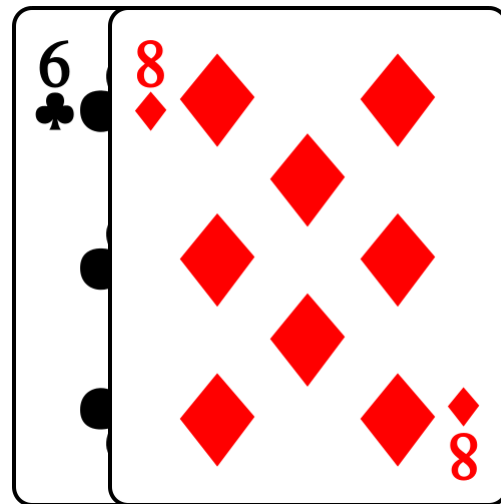
# Fusion de tableaux triés



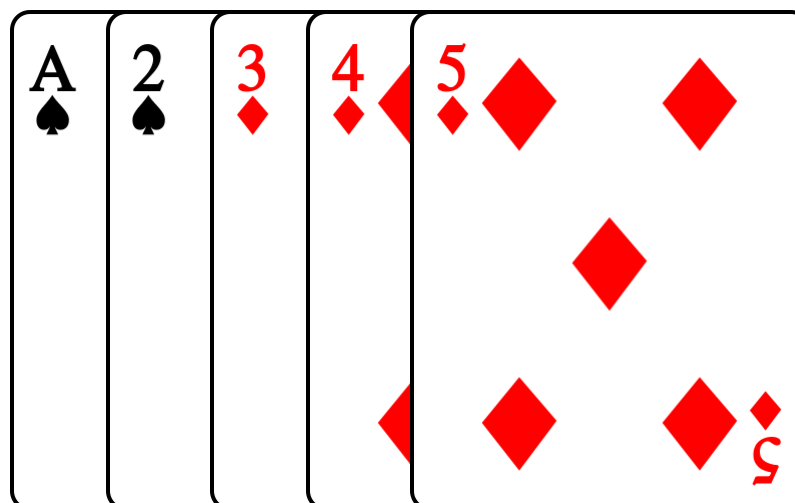
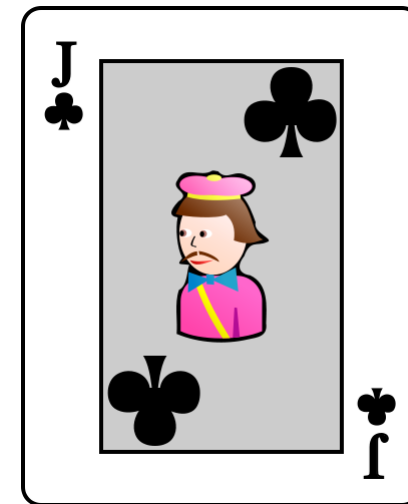
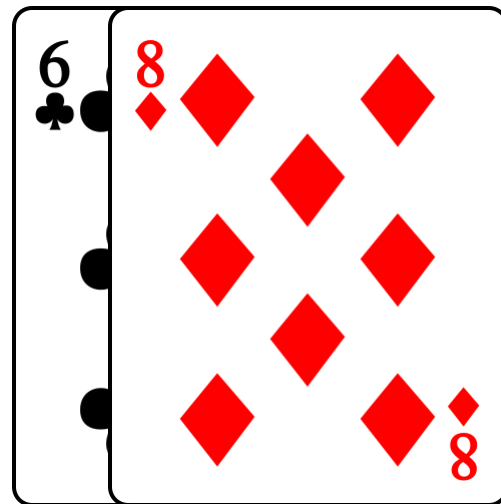
# Fusion de tableaux triés



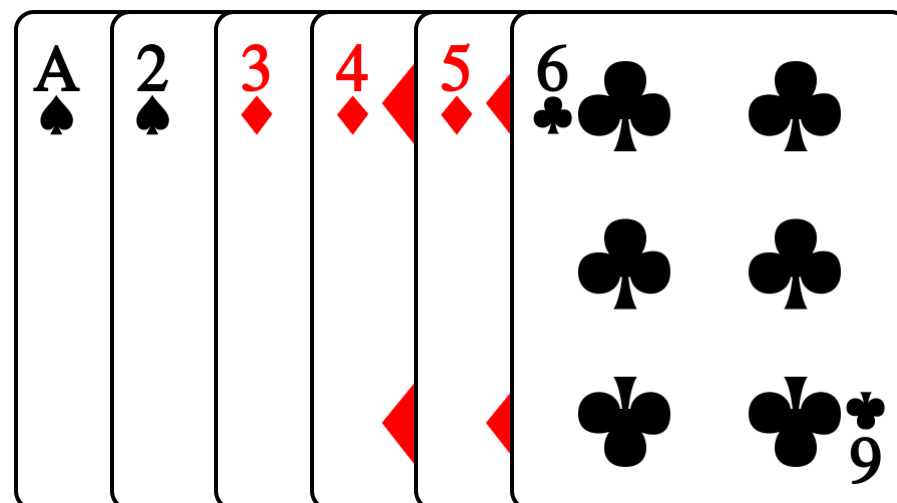
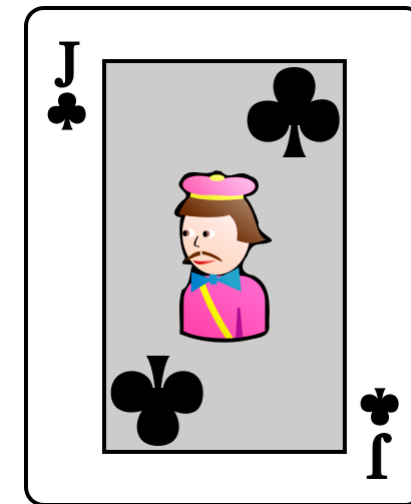
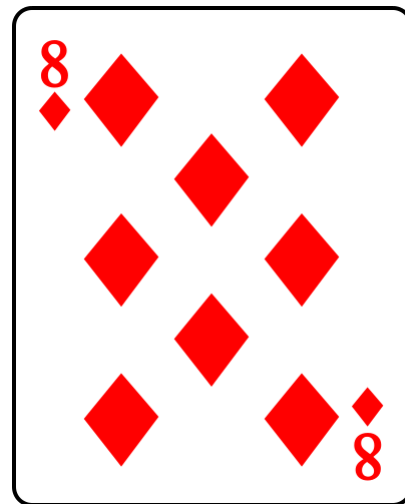
# Fusion de tableaux triés



# Fusion de tableaux triés

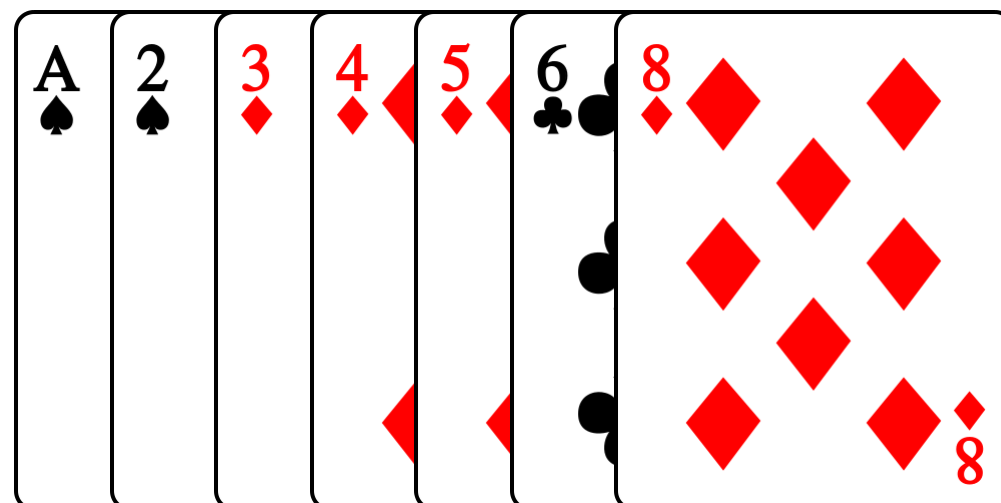
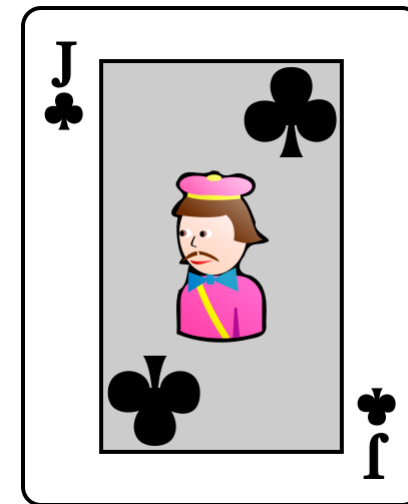


# Fusion de tableaux triés

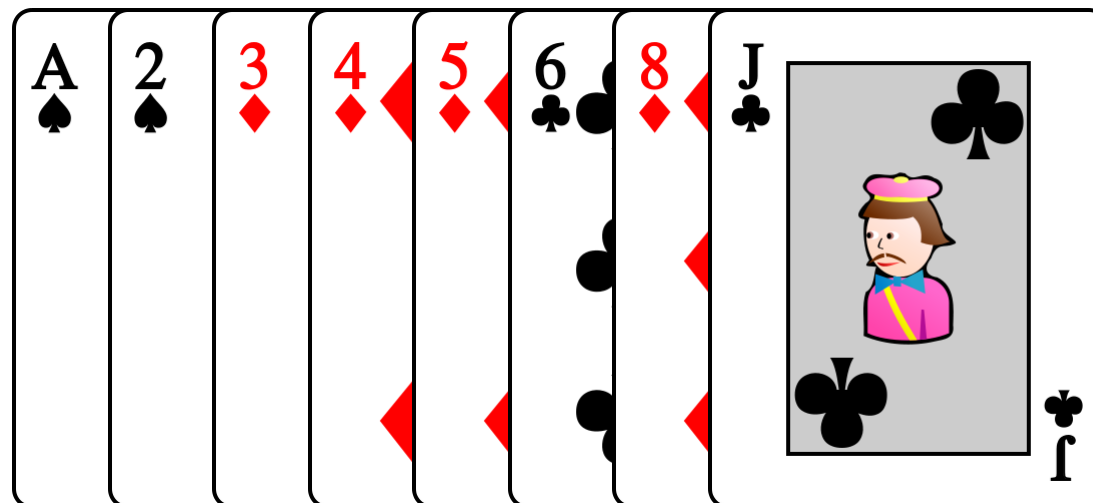




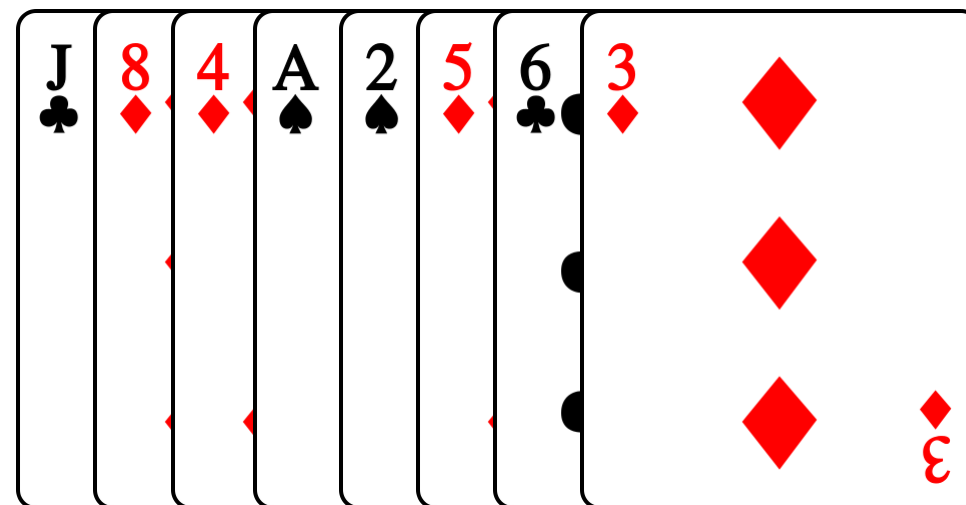
# Fusion de tableaux triés



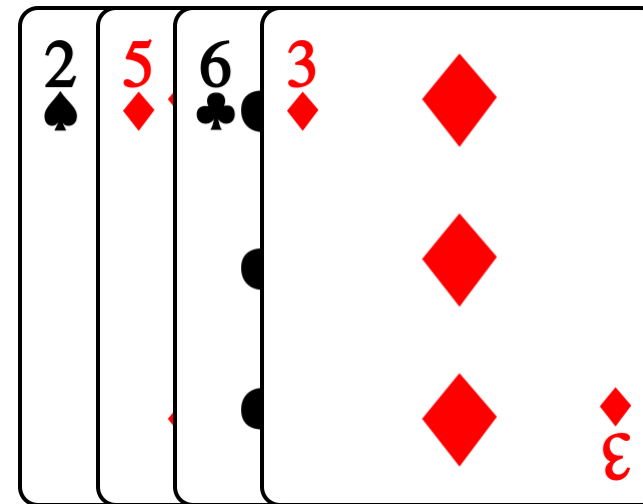
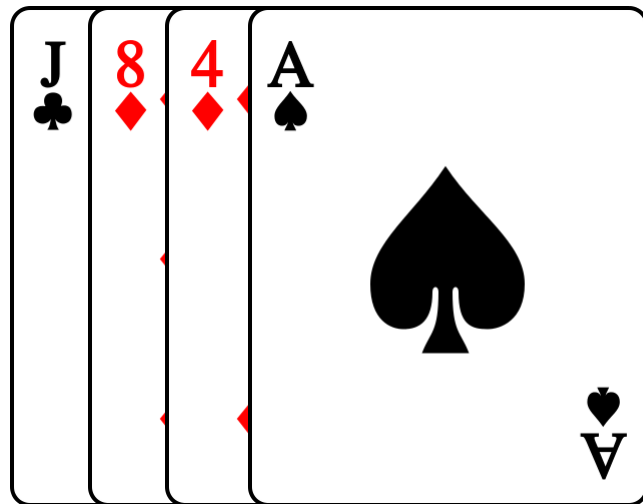
# Fusion de tableaux triés



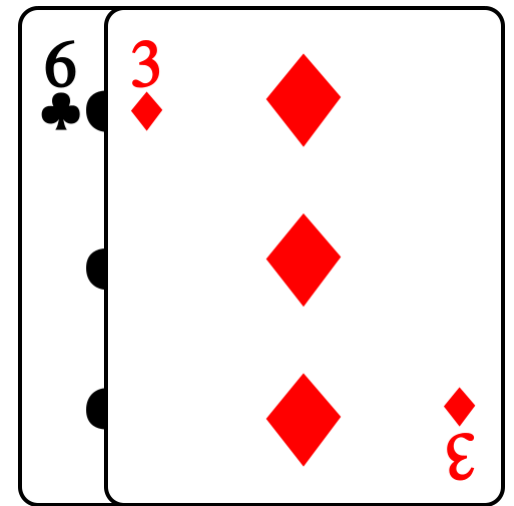
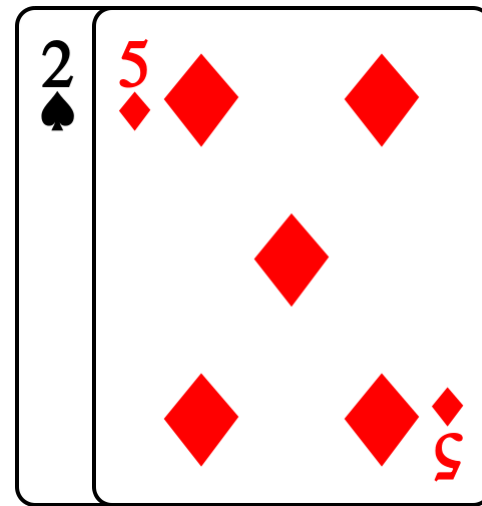
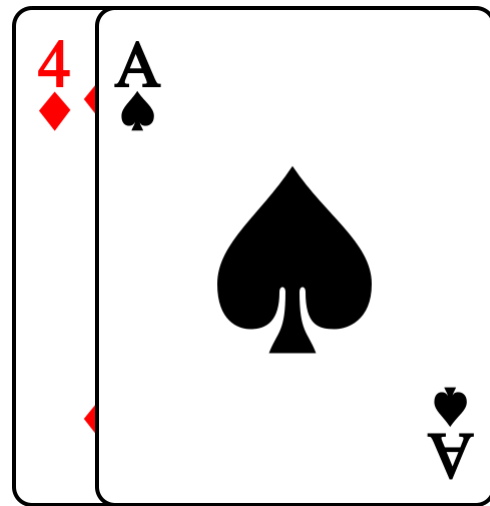
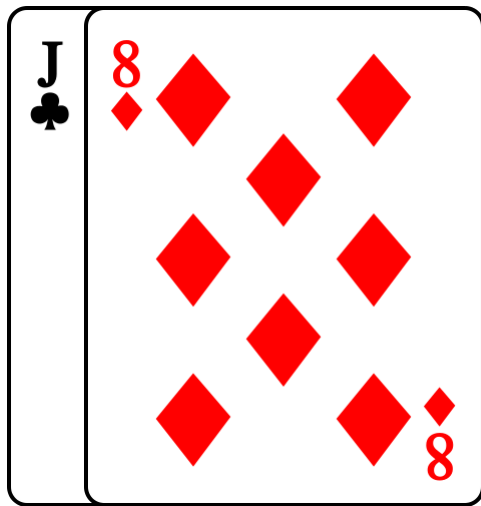
# Tri fusion



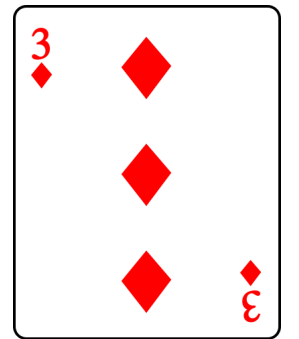
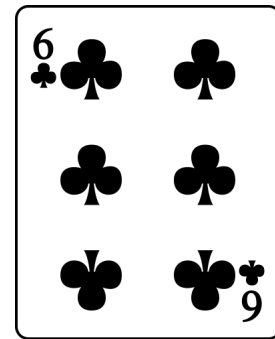
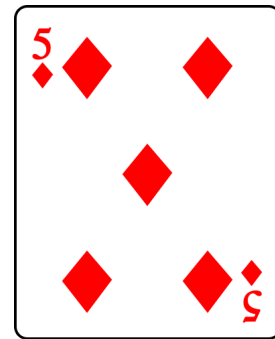
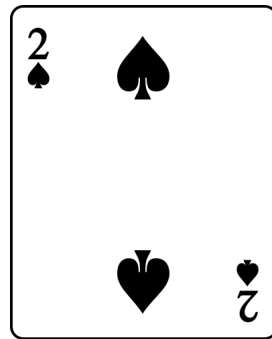
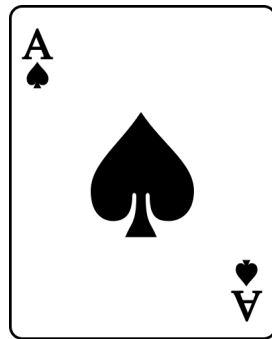
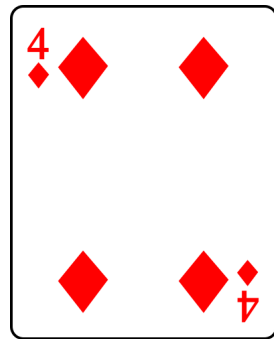
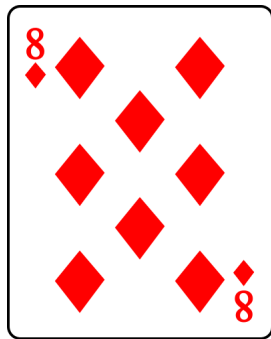
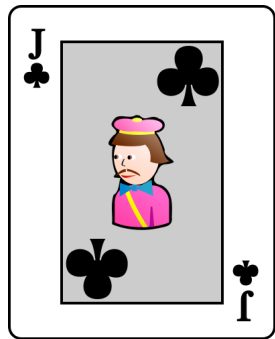
# Diviser



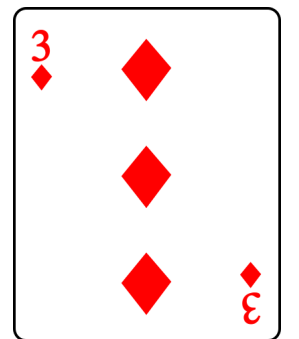
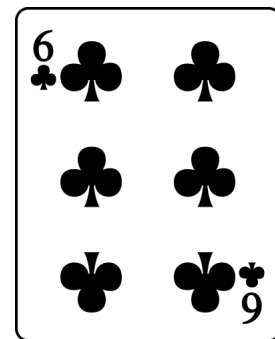
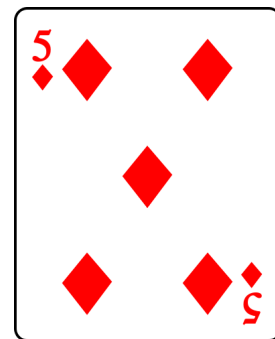
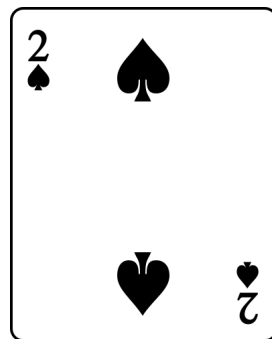
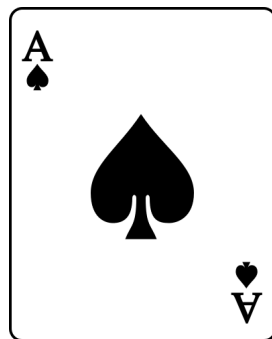
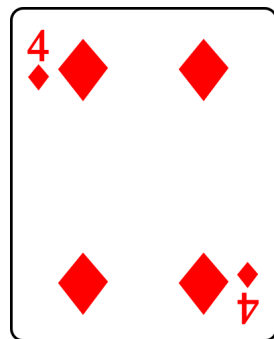
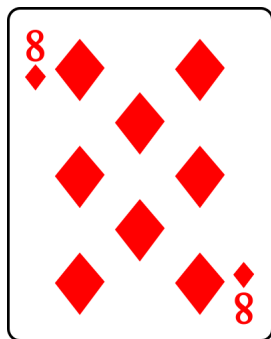
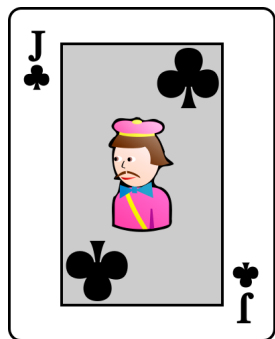
# Diviser



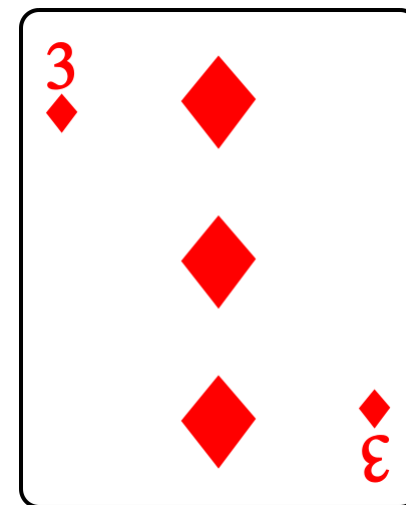
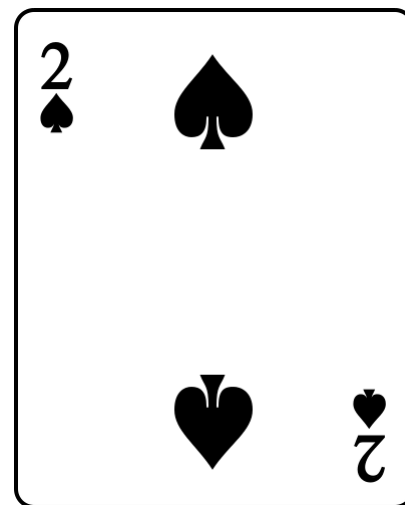
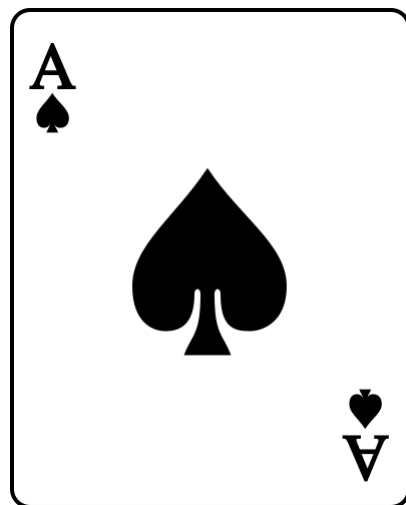
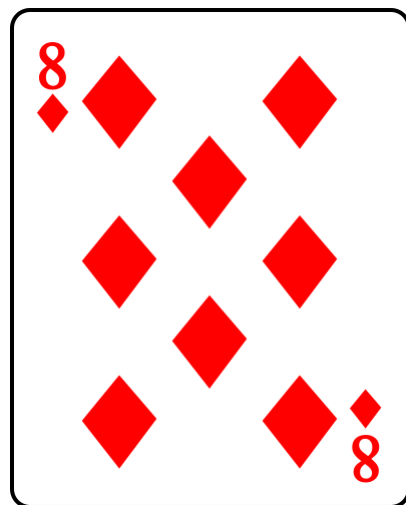
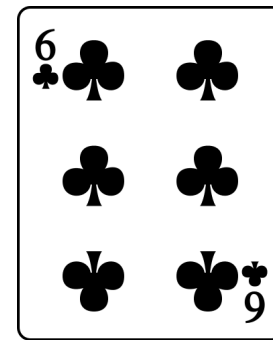
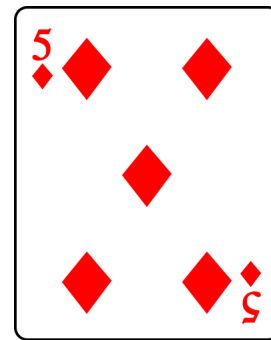
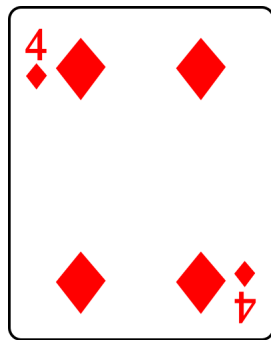
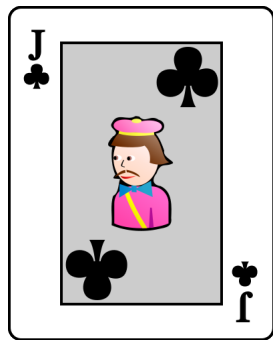
# Diviser



# Tous les jeux sont triés !

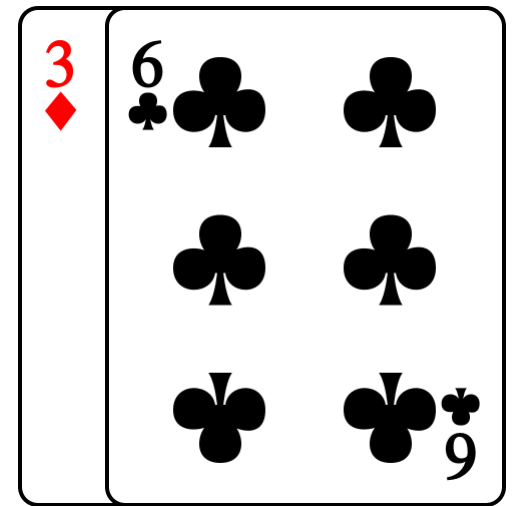
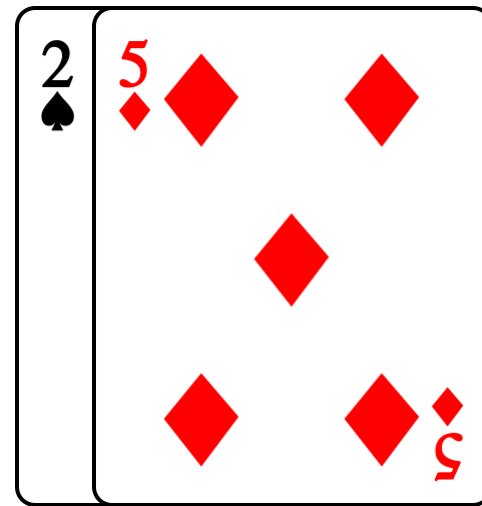
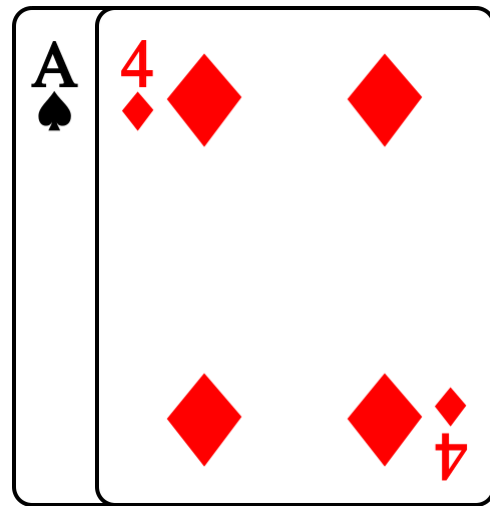
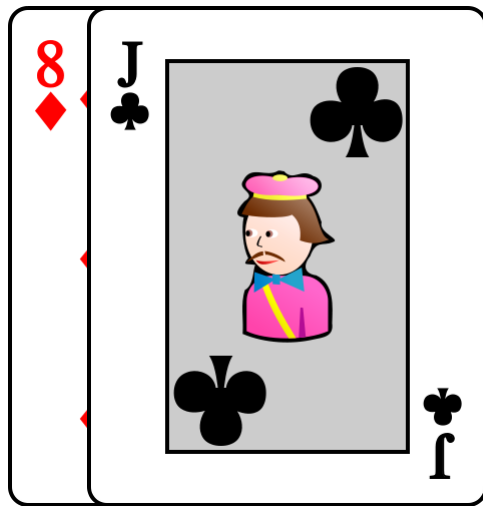


# Fusionner

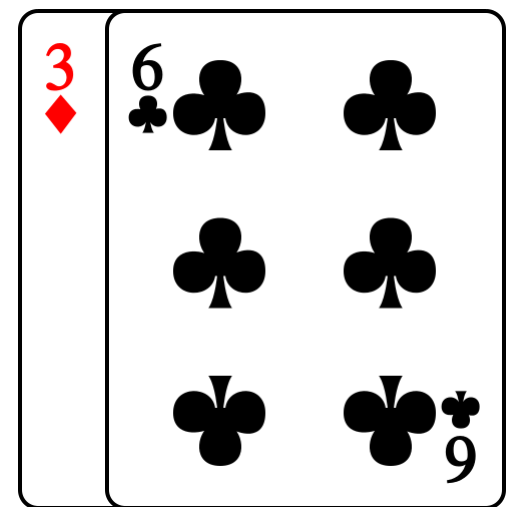
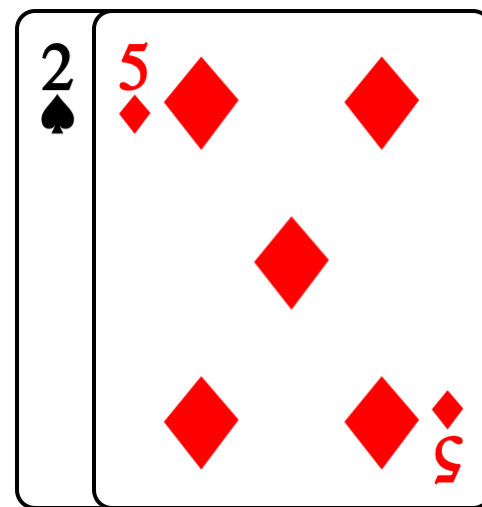
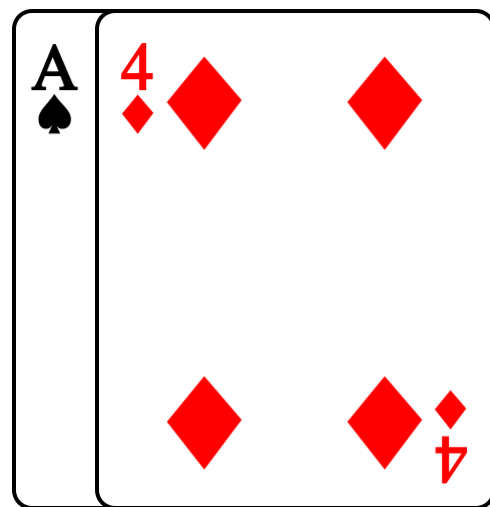
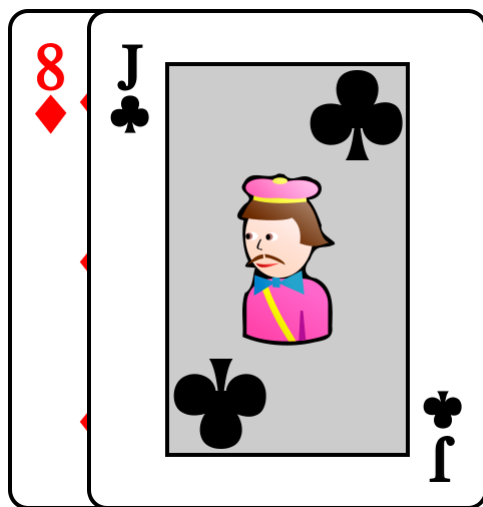




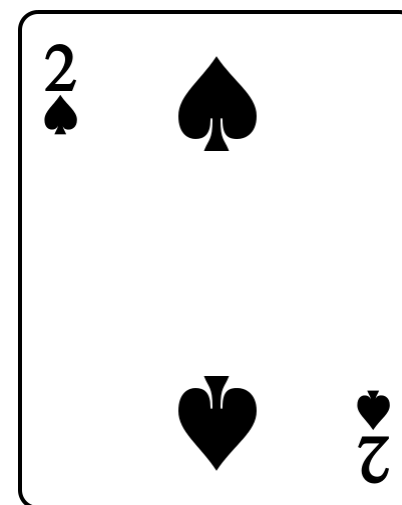
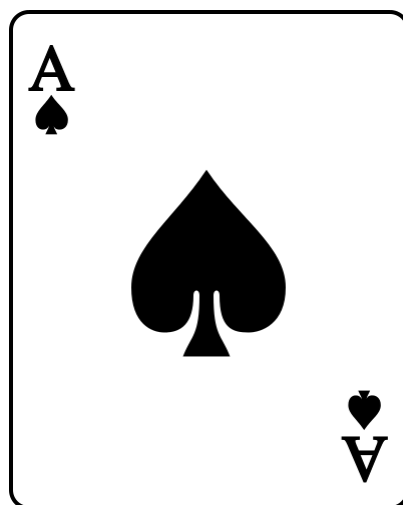
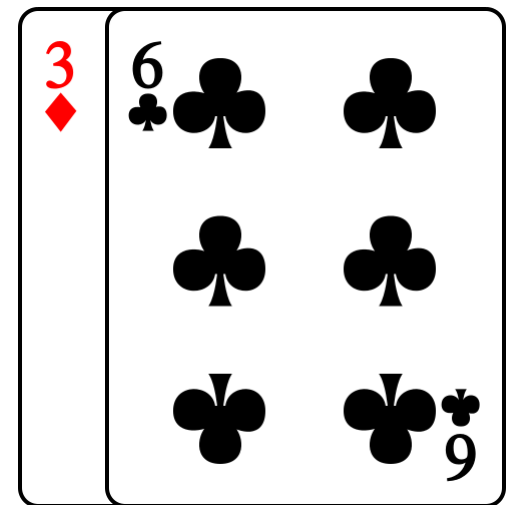
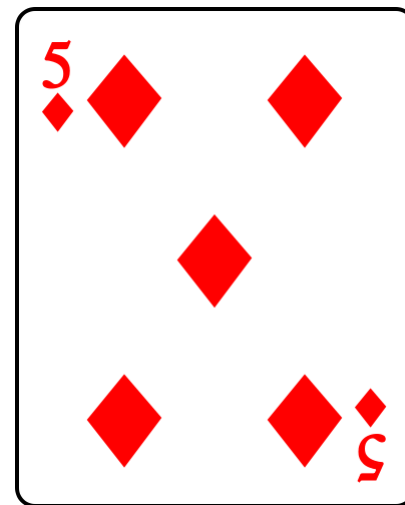
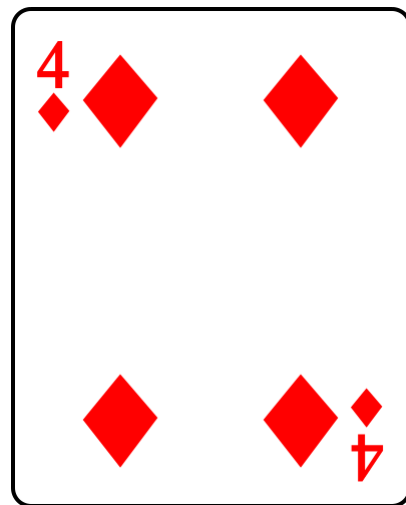
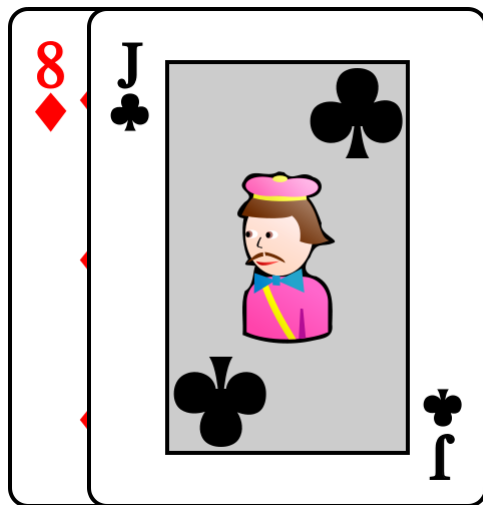
# Fusionner



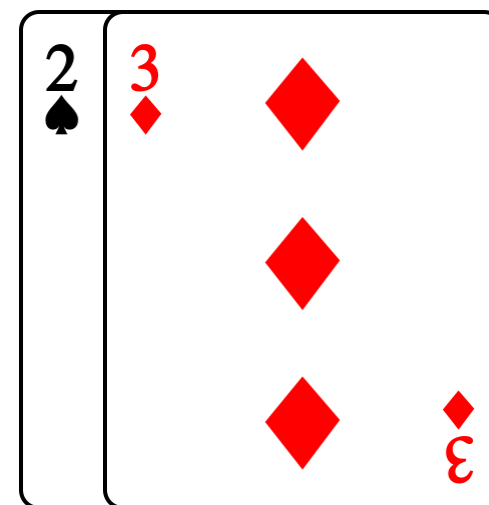
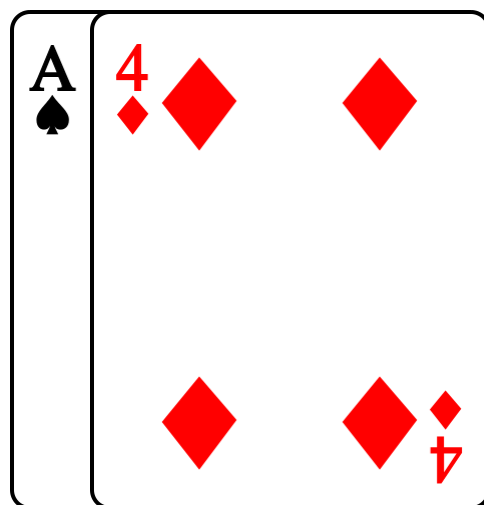
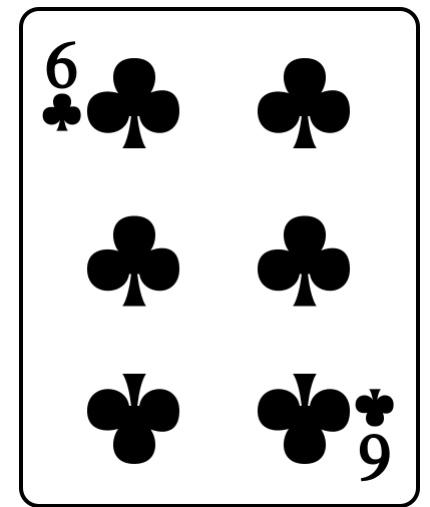
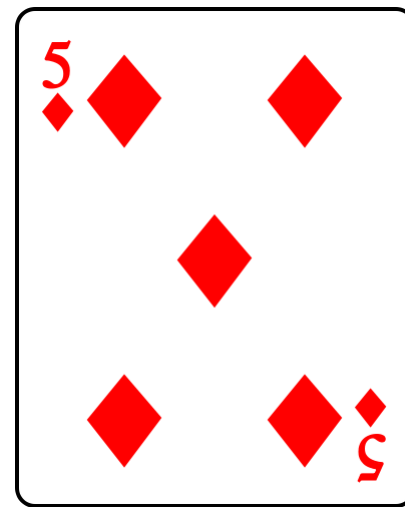
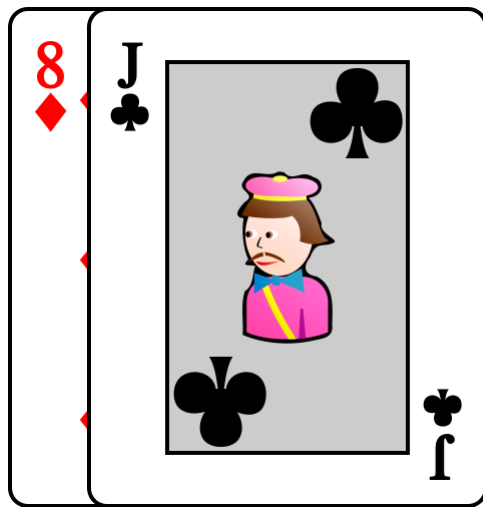
# Tous les jeux sont triés !



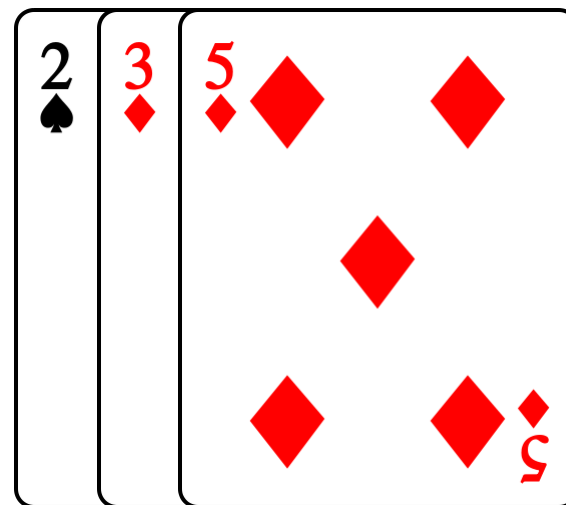
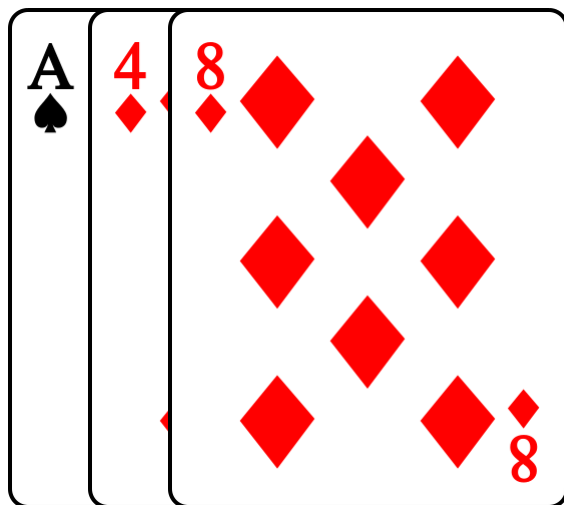
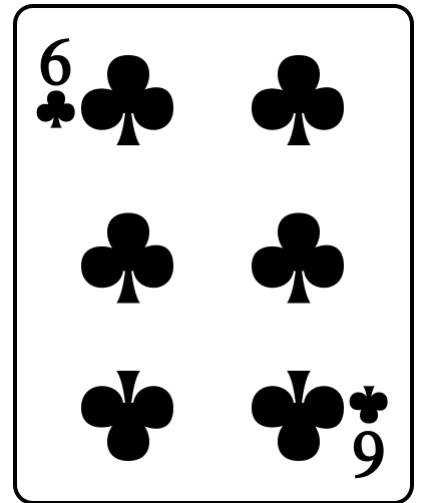
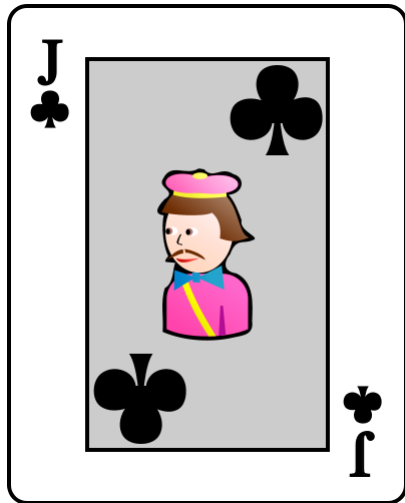
# Fusionner



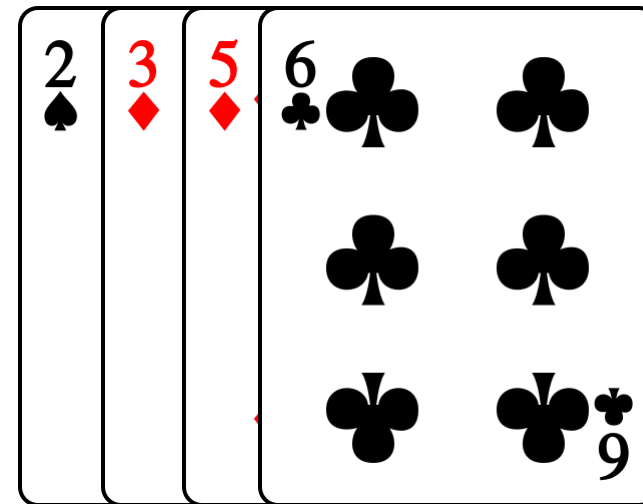
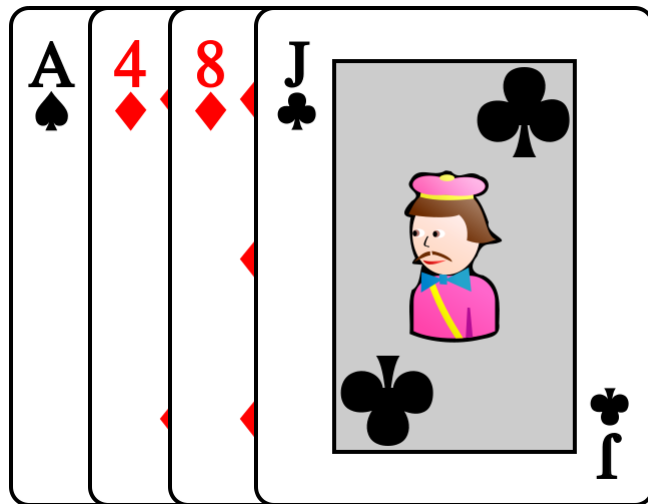
# Fusionner



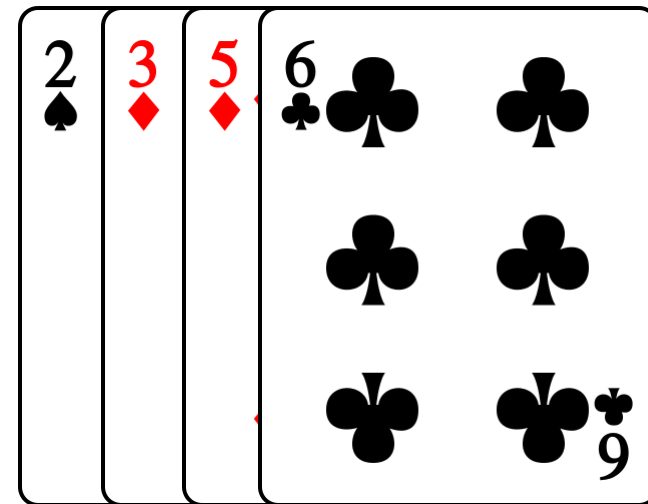
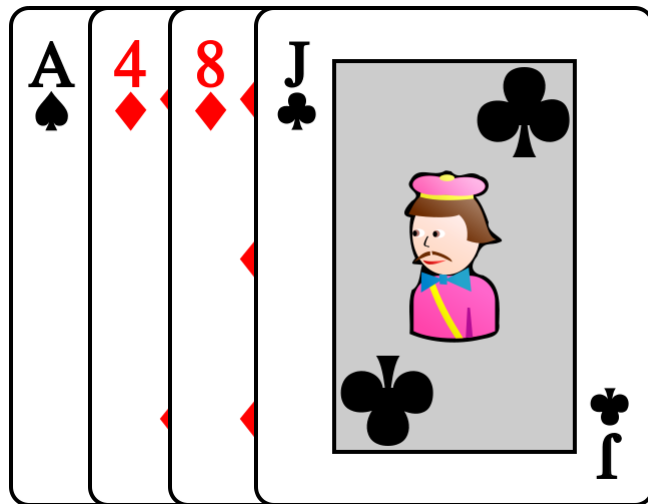
# Fusionner



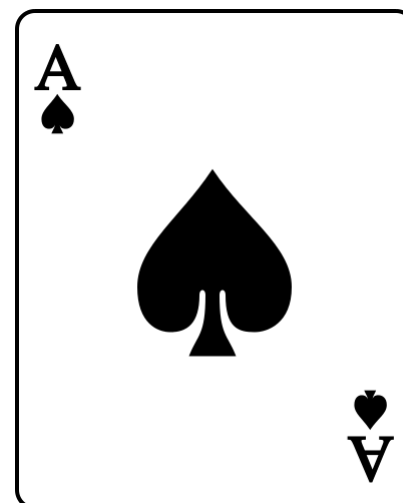
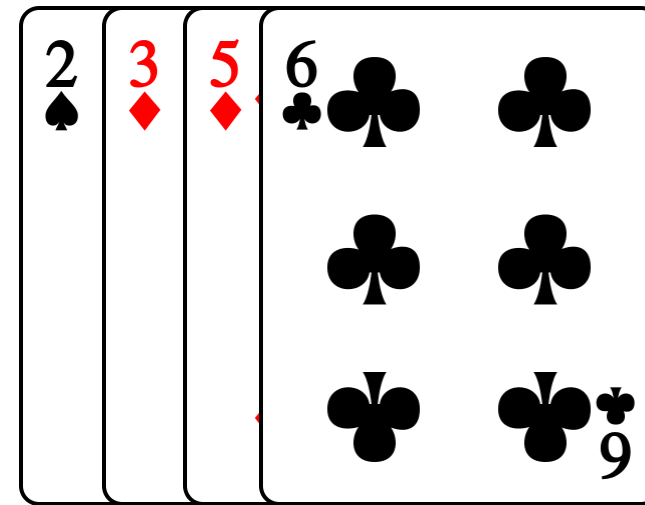
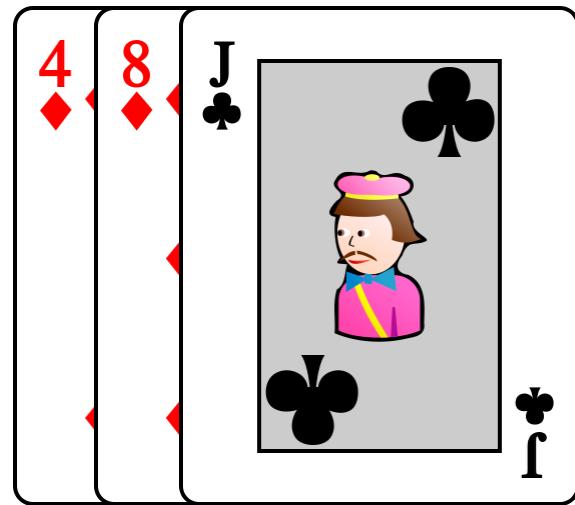
# Fusionner



# Tous les jeux sont triés !

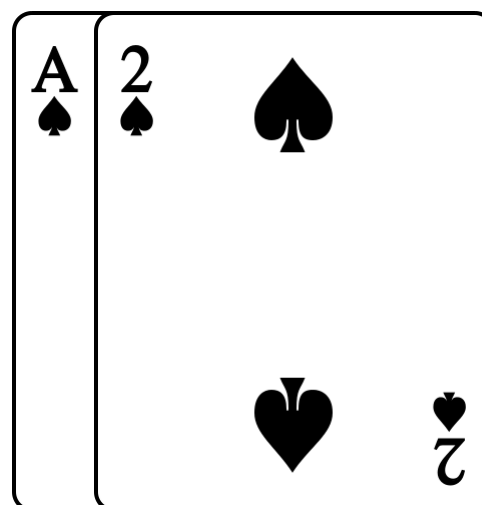
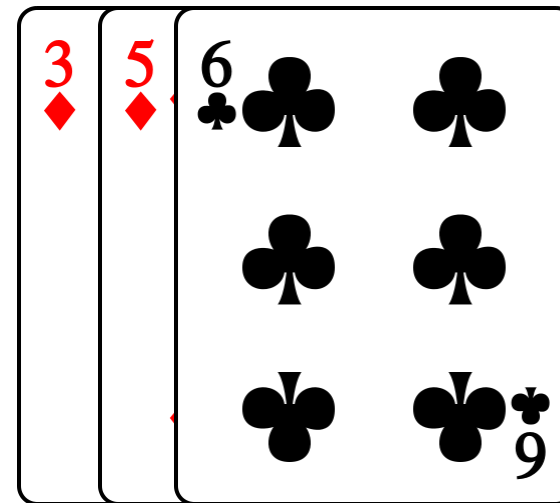
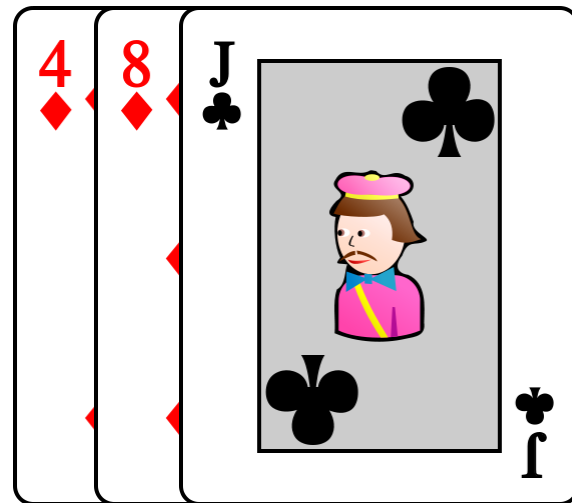


# Fusionner

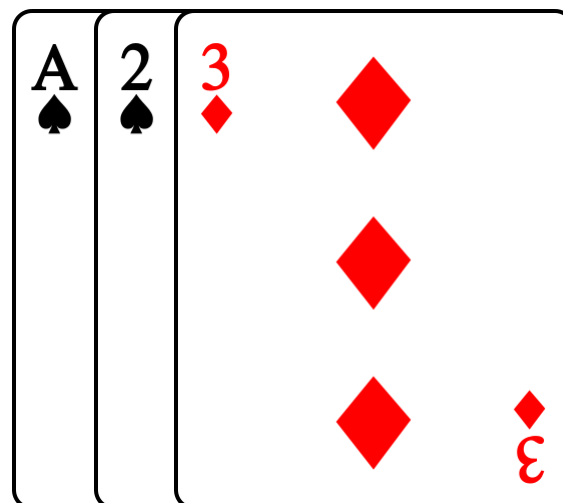
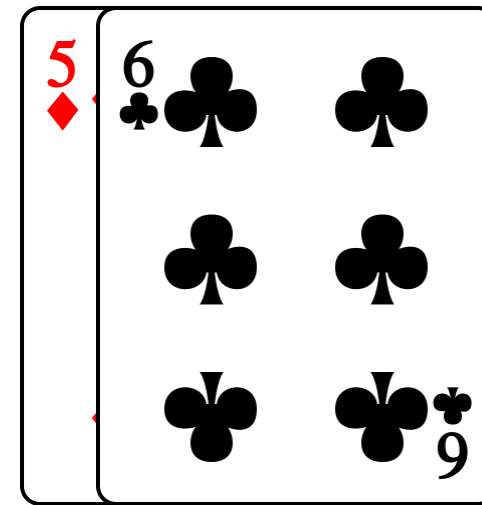
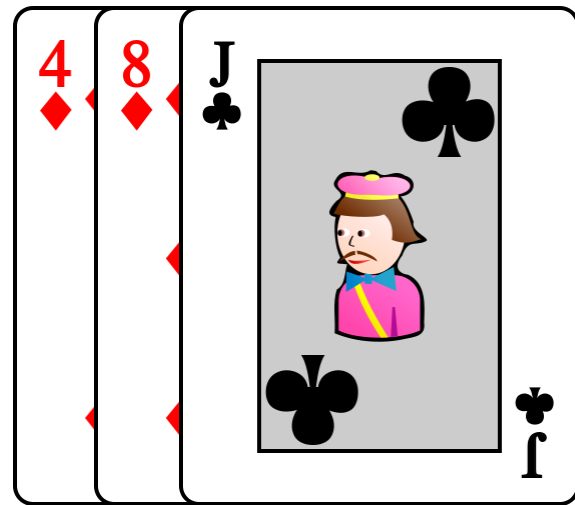




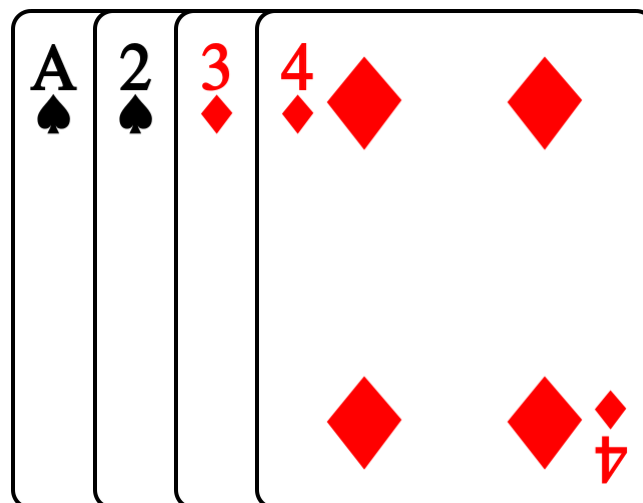
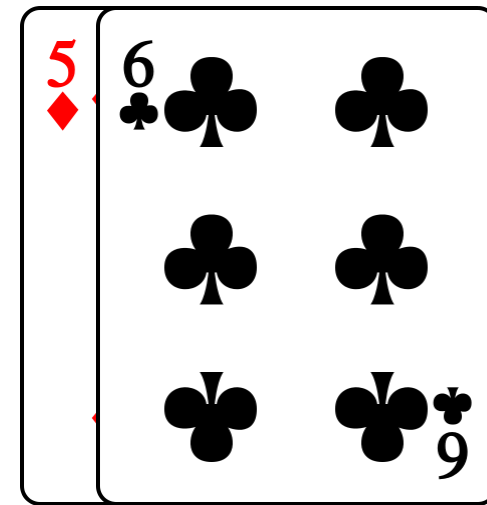
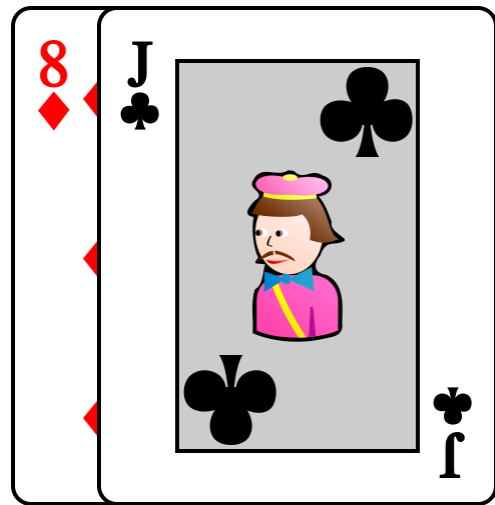
# Fusionner



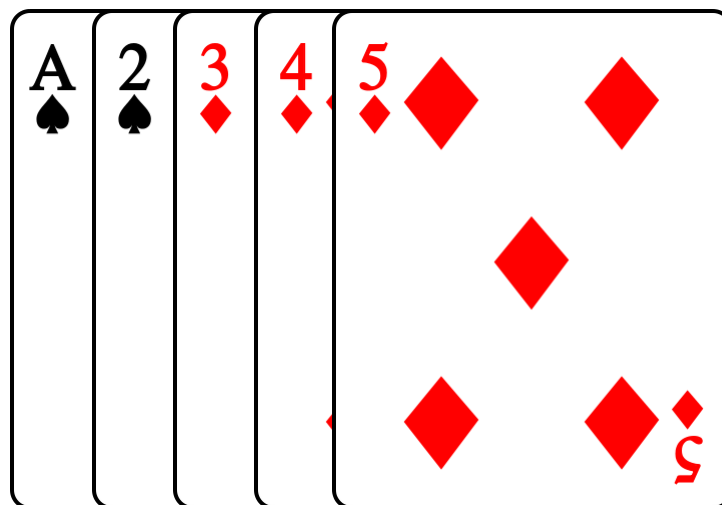
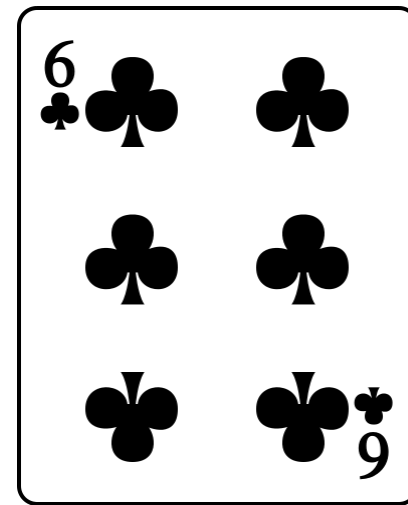
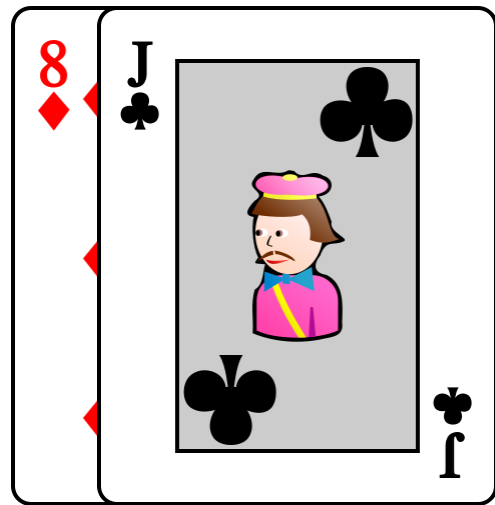
# Fusionner



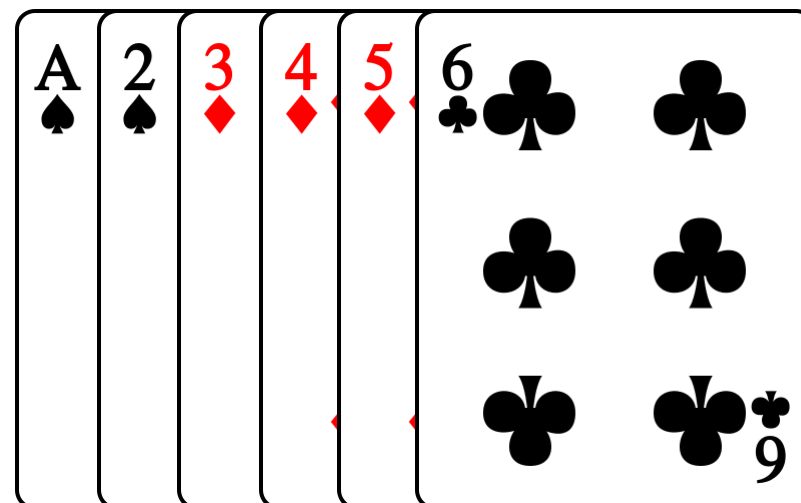
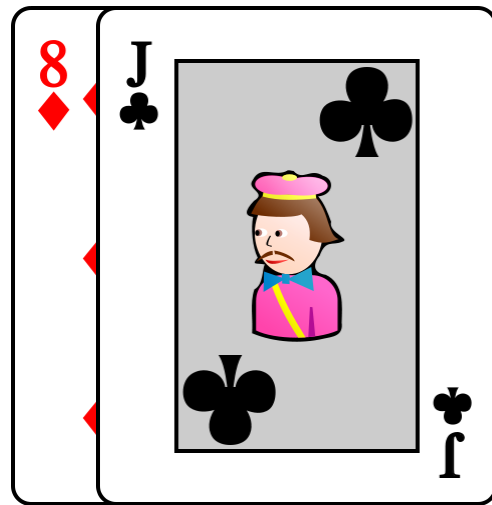
# Fusionner



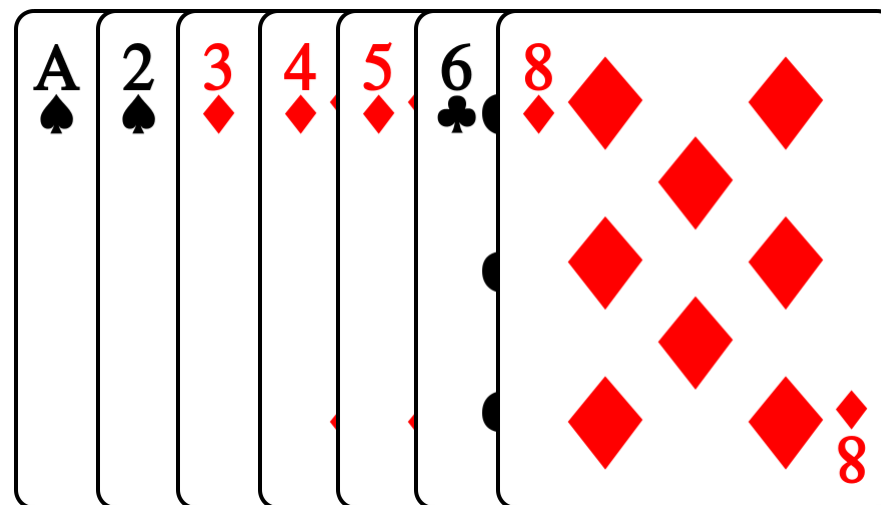
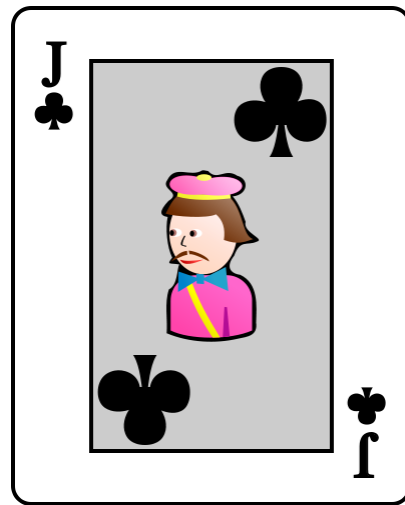
# Fusionner



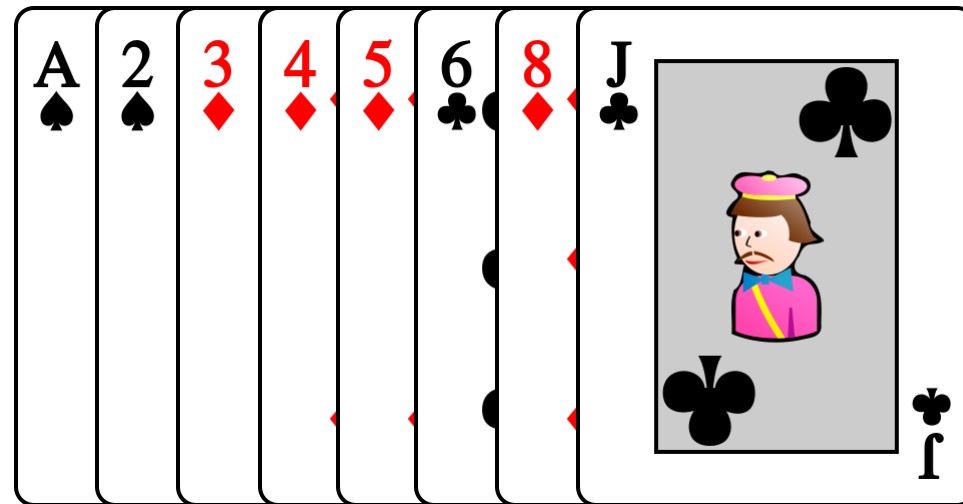
# Fusionner



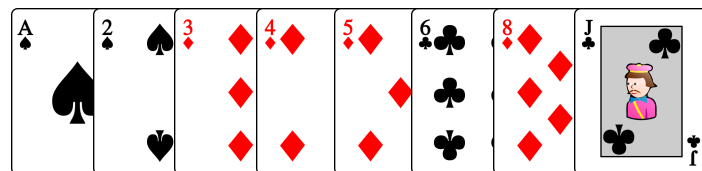
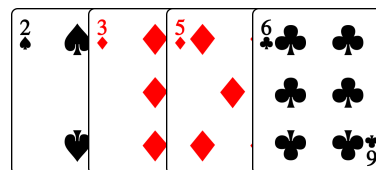
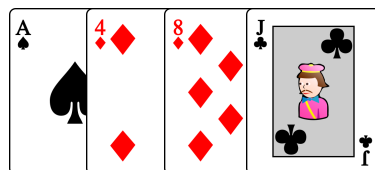
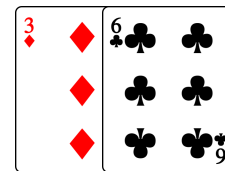
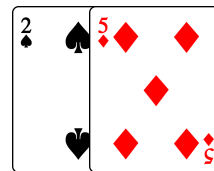
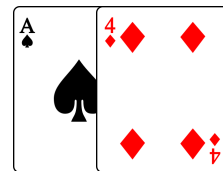
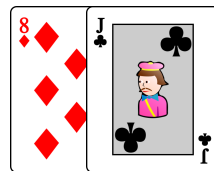
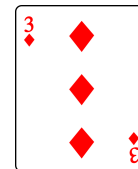
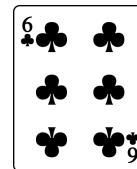
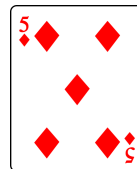
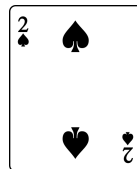
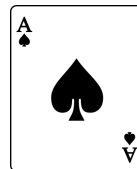
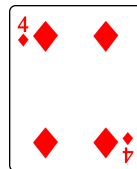
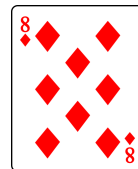
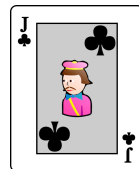
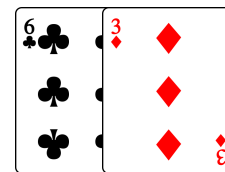
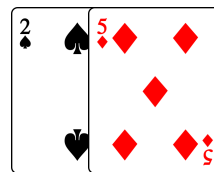
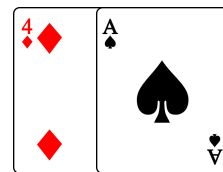
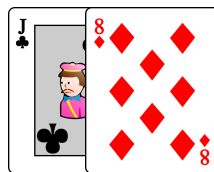
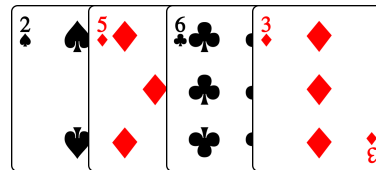
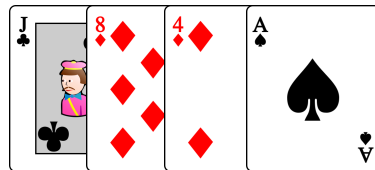
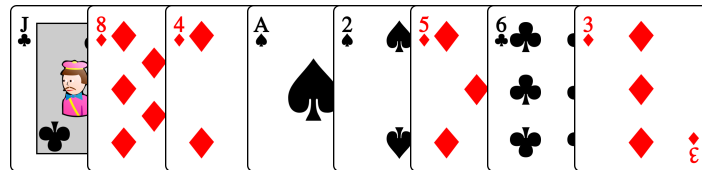
# Fusionner



# Le jeu est trié !

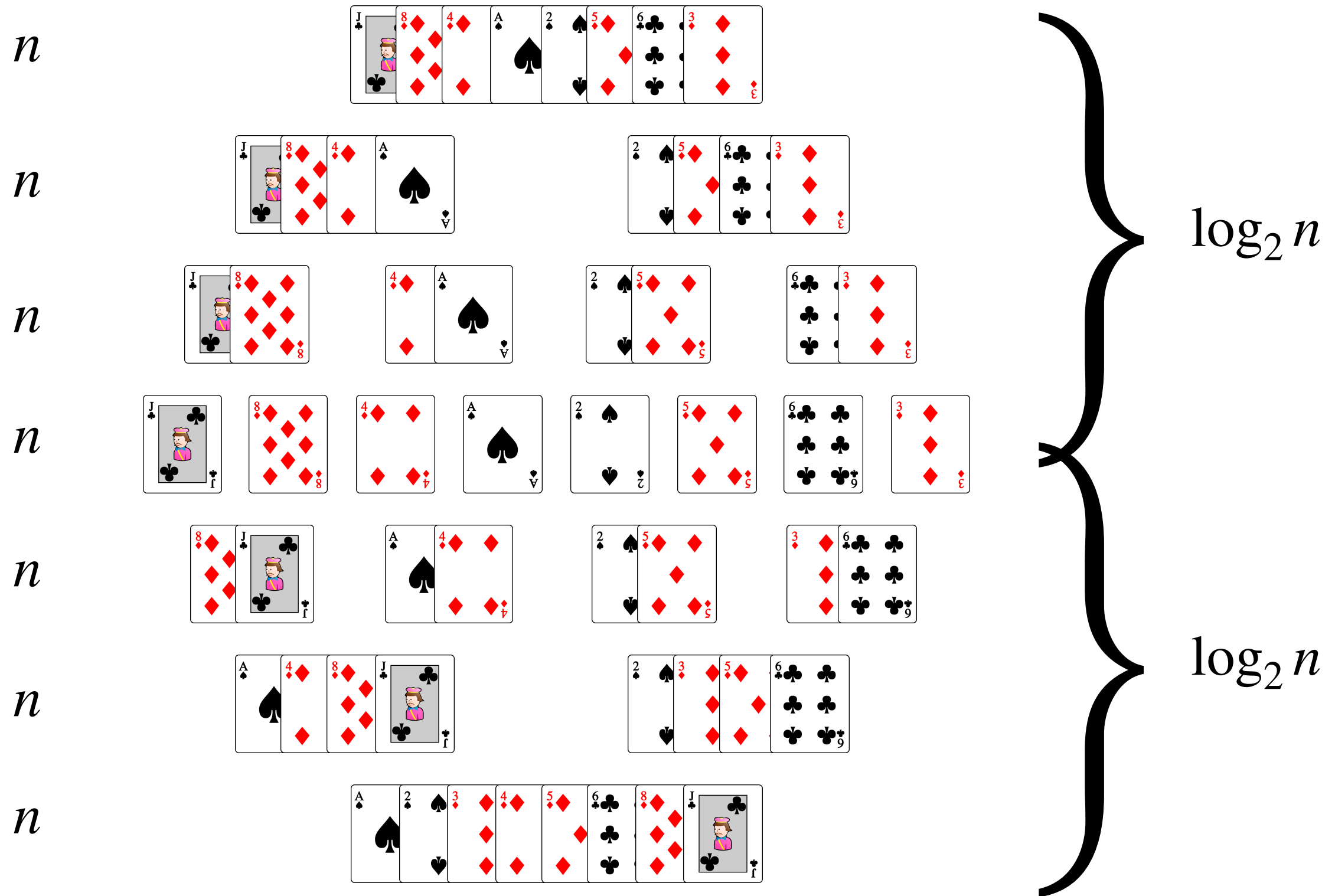


# Efficacité du tri fusion





# Efficacité du tri fusion



**La complexité  
du tri fusion est**

$$O(n \log_2 n)$$

# Fusion

```
fonction fusionner(A, B)
  n := longueur(A)
  m := longueur(B)
  C := tableau de longueur m + n
  i := j := k := 0
  tant que i < n et j < m faire
    si A[i] < B[j] alors
      C[k] := A[i]
      i := i + 1
    sinon
      C[k] := B[j]
      j := j + 1
    fin si
    k := k + 1
  fin tant que
  tant que i < n faire
    C[k] := A[i]
    i := i + 1
    k := k + 1
  fin tant que
  tant que j < m faire
    C[k] := B[j]
    j := j + 1
    k := k + 1
  fin tant que
  retourner C
fin fonction
```

encore d'éléments  
dans A et B

encore d'éléments  
dans A, mais B épuise

encore d'éléments  
dans B, mais A épuisé

A et B épuisés

# Fusion

**fonction** fusionner(A, B)

n := longueur(A)

m := longueur(B)

C := tableau de longueur m + n

i := j := k := 0

**tant que** i < n **et** j < m **faire**

**si** A[i] < B[j] **alors**

        C[k] := A[i]

        i := i + 1

**sinon**

        C[k] := B[j]

        j := j + 1

encore d'éléments  
dans A et B

**fonction fusionner(A, B)**

$n := \text{longueur}(A)$

$m := \text{longueur}(B)$

$C := \text{tableau de longueur } m + n$

$i := j := k := 0$

**tant que**  $i < n$  **et**  $j < m$  **faire**

**si**  $A[i] < B[j]$  **alors**

$C[k] := A[i]$

$i := i + 1$

**sinon**

$C[k] := B[j]$

$j := j + 1$

**fin si**

$k := k + 1$

**fin tant que**

**tant que**  $i < n$  **faire**

$C[k] := A[i]$

$i := i + 1$

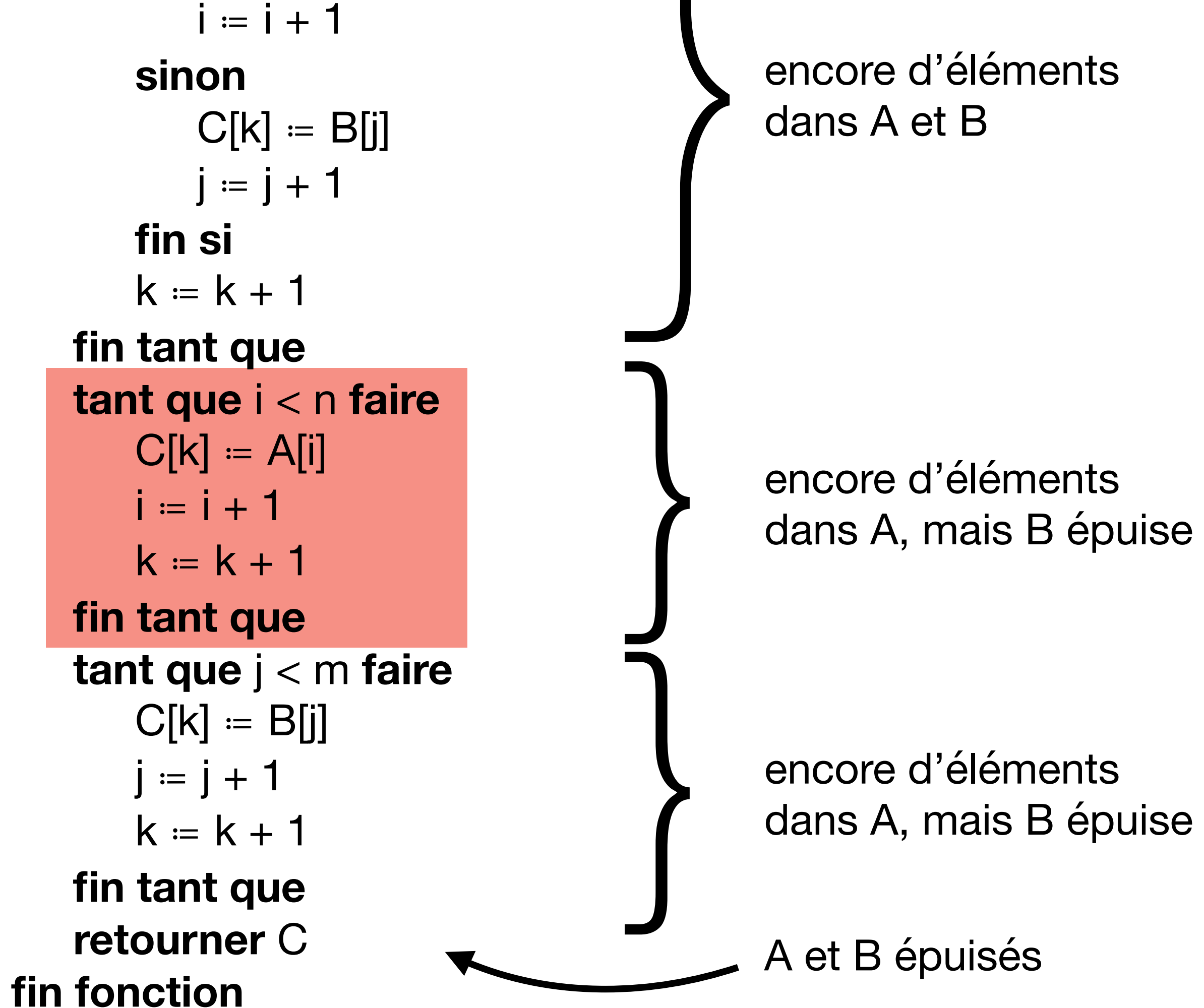
$k := k + 1$

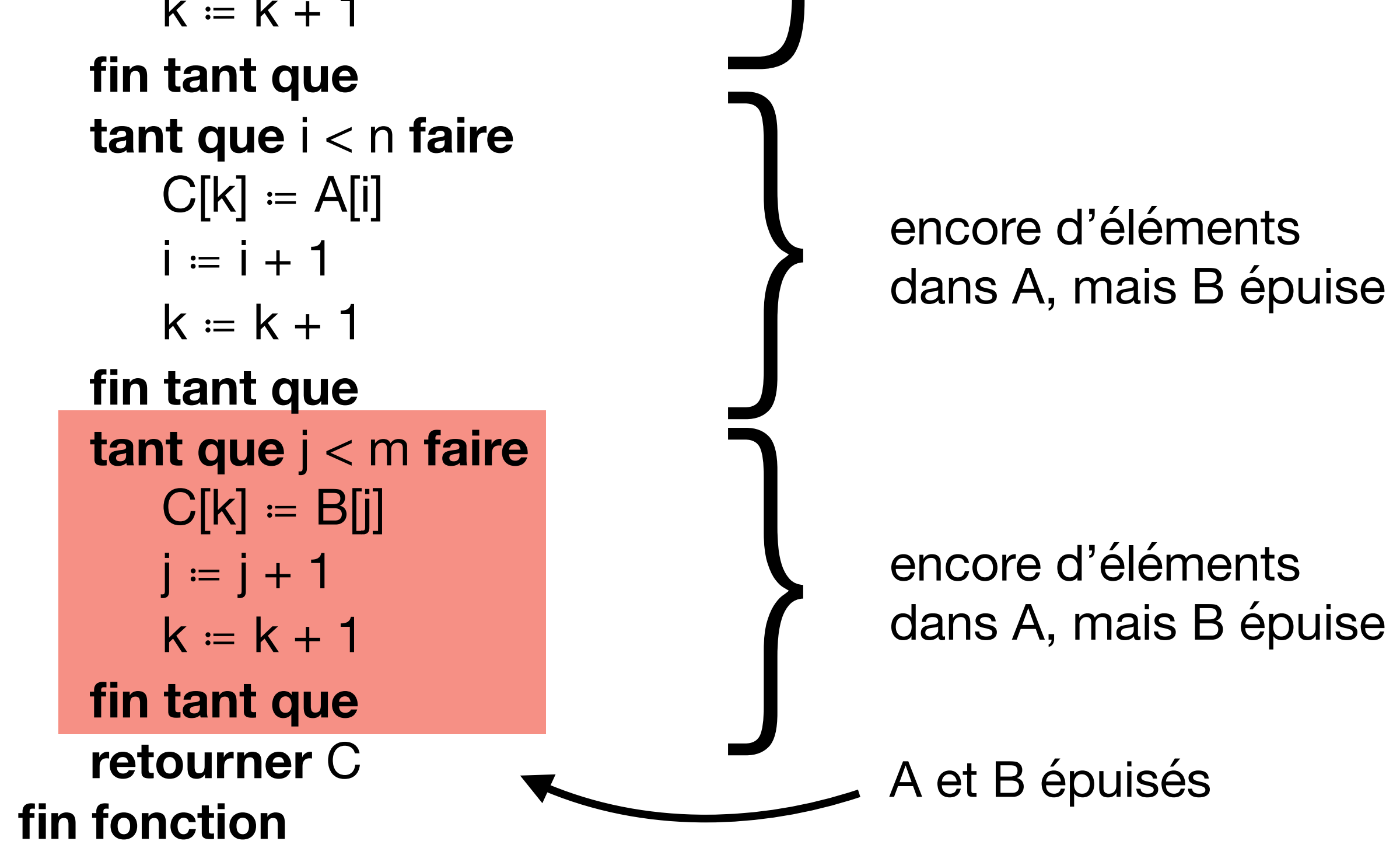
**fin tant que**

# Fusion

encore d'éléments  
dans A et B

encore d'éléments  
dans A, mais B épuise





**tant que  $i < n$  faire**

$C[k] := A[i]$

$i := i + 1$

$k := k + 1$

**fin tant que**

**tant que  $j < m$  faire**

$C[k] := B[j]$

$j := j + 1$

$k := k + 1$

**fin tant que**

**retourner C**  
**fin fonction**

encore d'éléments  
dans A, mais B épuise

encore d'éléments  
dans A, mais B épuise

A et B épuisés



# Fusion

**fonction** fusionner(A, B)

n := longueur(A)

m := longueur(B)

i := j := k := 0

**tant que** i < n **et** j < m **faire**

**si** A[i] < B[j] **alors**

    C[k] := A[i]

    i := i + 1

**sinon**

    C[k] := B[j]

    j := j + 1

**fin si**

  k := k + 1

**fin tant que**

**tant que** i < n **faire**

  C[k] := A[i]

  i := i + 1

  k := k + 1

**fin tant que**

**tant que** j < m **faire**

  C[k] := B[j]

  j := j + 1

  k := k + 1

**fin tant que**

**retourner** C

**fin fonction**

encore d'éléments  
dans A et B

encore d'éléments  
dans A, mais B épuise

encore d'éléments  
dans B, mais A épuisé

A et B épuisés

# Tri fusion

```
fonction tri-fusion(A)
  n := longueur(A)
  si n > 1 alors
    m :=  $\lfloor n \div 2 \rfloor$ 
    B := A[0, ..., m-1]
    C := A[m, ..., n-1]
    B' := tri-fusion(B)
    C' := tri-fusion(C)
    retourner fusionner(B', C')
  sinon
    retourner A
  fin si
fin fonction
```