

TD - Tas et File de Priorité

I Un tas de questions

II Compression de Huffman

1.

```
let rec read tree list =
  match tree with
  | F(a) -> a, list
  | N(g, d) -> let h, t = List.hd list, List.tl list in
                 if h = 0 then
                   read g t
                 else if h = 1 then
                   read d t
                 else failwith "Provided list does not correspond to a char"
;;
```
2.

```
let rec decode tree list =
  if list = [] then [] else
  let char, remainder = read tree list in
  char::decode tree remainder
;;
```
- 3.
- 4.

III Arbretas

1.

```
let swap tableau i j =
  let temp = tableau.(i) in
  begin
  tableau.(i) <- tableau.(j);
  tableau.(j) <- temp ;
  end ;;
```
2. TODO
3. Montrons le par récurrence sur le nombre d'élément de l'arbre. Notons \mathcal{P}_k , l'assertion "Pour k couples distincts, il existe un unique arbretas les contenant".
 - Initialisation : si $k = 0$, alors l'arbre vide convient.
 - Soit $k \in \mathbb{N}^*$, supposons \mathcal{P}_{k-1} : Soit k couples distincts, on les note $E = (e_1, p_1), \dots, (e_k, p_k)$ On suppose sans perdre de généralité que les p_i sont triés dans l'ordre décroissant. Un arbretas contenant E est donc de la forme $N(r, g, d)$ où $r = (e_k, p_k)$ et g, d sont des arbretas contenant au plus $k - 1$ éléments donc sont uniques d'après \mathcal{P}_k .

D'où le résultat.

4.

```
let rotd abr =
  match abr with
  | N(r, N(gr, gg, gd), d) -> N(gr, gg, N(r, gd, d))
  | _ -> abr
```
5. TODO
6.

```
let prio = function
  | V -> max_int
  | N(r, g, d) -> r
```

```

7. let rec add abr (element, priorite) =
  match abr with
  | V -> N((element, priorite), V, V)
  | N((element_r, priorite_r), g, d) when element < element_r ->
    if priorite < priorite_r then
      rotd (N((element_r, priorite_r), add g (element, priorite), d))
    else
      N((element_r, priorite_r), add g (element, priorite), d)
  | N((element_r, priorite_r), g, d) ->
    if priorite < priorite_r then
      rotg (N((element_r, priorite_r), g, add d (element, priorite)))
    else
      N((element_r, priorite_r), g, add d (element, priorite))
;;

```

8. TODO