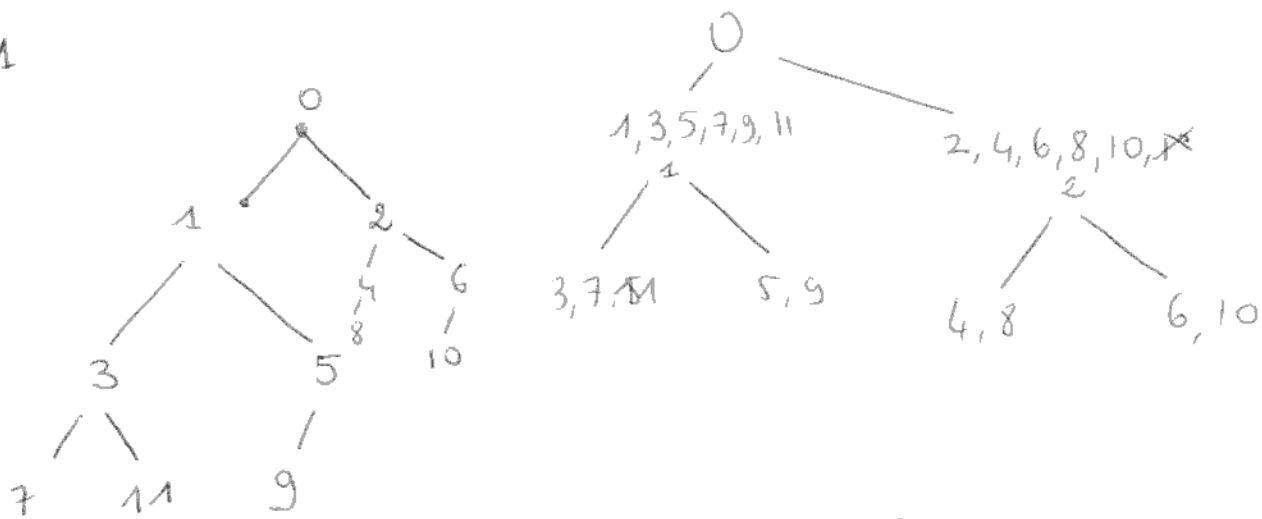


2.1



2.2. let get i a = match i with
 $10, E \rightarrow \text{fail with}$
 $10, N(g, x, d) \rightarrow x$
 $| i, E \rightarrow \text{fail with}$
 $| i, N(g, x, d) \text{ when } i \bmod 2 = 0 \rightarrow \text{get } (i/2) \text{ d}$
 $| i, N(g, x, d) \rightarrow \text{get } (i/2) \text{ g}$

2.3. On montre par récurrence sur $n \in \mathbb{N}$, qu'il existe une unique structure de tableau flexible pour chaque n .

Puis on montre par réc sur $n \in \mathbb{N}^*$ que ce tableau est de hauteur h avec $2^h \leq |\text{arbre}| = n$

• Si $n = 1$ alors $h = 0$ et c'est bon

• Si $n = 2i + 1$  $h = 1 + \max(h_i, h_{i+1})$
 avec h_i
 $2 \leq i$

donc $2^h = 2^{1+h_i} \leq 2i \leq n$

• Si $n = 2i$  $h = 1 + \max(h_i, h_{i-1})$
 avec $2^{h_i} \leq i$ et $2^{h_{i-1}} \leq i-1$

donc: $2^h = 2 \times 2^{\max(h_i, h_{i-1})} \leq 2i = n$

2.4.

$$C_1 = 1$$

$$C_0 = 1$$

$$C_n = C_{n/2} + O(1)$$

ou

$$C_h = C_{h-1} + O(1)$$

$$\Rightarrow C_h = O(h) = O(\log_2(n)) \text{ par 2.3.}$$

2.5. On montre le résultat par rec sur $n \in \mathbb{N}^*$.

si $n = 1$ ok $\rightarrow$ cas de base.

si $n = 2k$ est pair alors $|l| = k$ et $|r| = k-1$
et le dernier elt est en posit° $2k-1$ c'est donc le dernier
elt de l qui est de taille $k = n/2$ - ok par HR.

si $n = 2k+1$ est impair $|l| = |r| = k$ et le
dernier elt est en posit° $2k$ c'ad dernier de r de taille
 $k = \lfloor n/2 \rfloor$ ok par HR

2.6.

let rec snoc n t x = match t with

$$|E \rightarrow N(E, x, E)$$

$$|N(l, x, r) \text{ when } n \bmod 2 = 0 \rightarrow$$

$$N(l, x, \text{snoc}((n/2)-1) r \ x)$$

$$|N(l, x, r) \rightarrow N(\text{snoc}(n/2) l \ x, x, r)$$

2.7. On a besoin d'une fct calculant la racine.

let root a = match a with

$$|E \rightarrow \text{failwith}$$

$$|N(-, x, -) \rightarrow x$$

let tail a = match a with

$$|N(E, -, E) \rightarrow E$$

$$|N(g, x, d) \rightarrow N(d, \text{root } g, \text{tail } g)$$

$$|E \rightarrow \text{failwith.}$$

Tout d'abord $\hat{c} \quad |d| \leq |g| \leq |d|+1$

$$\text{alors } |d|-1 \leq |\text{tail } g| \leq |d|$$

et l'arbre renvoyé (si on suppose que tail renvoie ce que'il faut sur g) est bien flexible et l'inverse permet de mettre l'elt d'indice 1 en racine, ceux d'indices 2, 4, ... de la sag et ceux d'indices 3, 5... de la sous arbre dt.

A chaque appel rec la hauteur diminue donc

$$Ch = O(h) = O(\log n)$$

2.8.

let cons elt a = match a with

$$|E \rightarrow N(E, \text{elt}, E)$$

$$|N(g, x, d) \rightarrow N(\text{cons } \overset{x}{\cancel{d}}, \text{elt}, g)$$

2.9.