

# Randomised Algorithms

## Winter term 2022/2023, Exercise Sheet No. 7

### Authors:

Ben Ayad, Mohamed Ayoub  
Kamzon, Nouredine

December 4, 2022

#### Exercise 1.

(a)

$$\text{We have for } t \geq 1: \quad X_{t+1} = \begin{cases} X_t + \lfloor X_t/2 \rfloor & \text{with probability } 18/37 \\ X_t - \lfloor X_t/2 \rfloor & \text{Otherwise} \end{cases}$$

Technically speaking, we can never reach the state 0, as the player gets stuck when  $X_t = 1$ . We'll define  $Y_t = X_t - 1$ . We have:  $Y_{t+1} = \begin{cases} Y_t + \lfloor \frac{Y_t+1}{2} \rfloor & \text{with probability } 18/37 \\ Y_t - \lfloor \frac{Y_t+1}{2} \rfloor & \text{Otherwise} \end{cases}$

Which yields, for  $s \geq 1$ :

$$\begin{aligned} \mathbb{E}[Y_t - Y_{t+1} | Y_t = s] &= s - \mathbb{E}[Y_{t+1} | Y_t = s] \\ &= s - \left( \frac{18}{37} \left( s + \lfloor \frac{s+1}{2} \rfloor \right) + \frac{19}{37} \left( s - \lfloor \frac{s+1}{2} \rfloor \right) \right) \\ &= \frac{1}{37} \lfloor \frac{s+1}{2} \rfloor \end{aligned}$$

And hence:  $\frac{s+1}{74} - \frac{1}{74} < \mathbb{E}[Y_t - Y_{t+1} | Y_t = s] \leq \frac{s+1}{74} \implies \frac{s}{74} < \mathbb{E}[Y_t - Y_{t+1} | Y_t = s] \leq \frac{s+1}{74}$

As  $h : s \mapsto \frac{s}{74}$  is monotonically increasing, using the Variable Drift Theorem would lead us to:

$$\mathbb{E}[T | Y_0 = X_0 - 1] =$$

(b)

#### Exercise 2.

(a) The worst case is to have all pairs  $\{a_i, a_j\}$  s.t:  $i < j$  not ordered, in this case  $INV(S_0) = \binom{n}{2} = \frac{n(n-1)}{2}$ . We can achieve so, by taking a strictly ordered list, and inverse the order.

(b) We let  $S_t = [a_1, \dots, a_n]$  and  $S_{t+1} = [b_1, \dots, b_n]$ . We have,  $b_i = a_j$  and  $b_j = a_i$ . We need to prove that:

$$INV(S_{t+1}) \subsetneq INV(S_t)$$

For the strictness, obviously,  $\{a_i, a_j\} \in INV(S_t)$  but  $\{a_i, a_j\} = \{b_j, b_i\} \notin INV(S_{t+1})$ , now we need to prove the inclusion:

Let  $\{b_k, b_l\} \in INV(S_{t+1})$  s.t:  $k < l$ : we have ofc  $b_l < b_k$

if  $k \neq i$  and  $l \neq j$ , then  $(b_k, b_l) = (a_k, a_l)$  and hence,  $\{a_k, a_l\} \in INV(S_t)$ .

If  $k = i$  and  $l < j$ :

If  $k = i$  and  $i < j < l$ :

If  $k < i$  and  $j = l$ :

If  $i < k < l$  and  $j = l$ :

**Exercise 3.**

**Exercise 4.**

Let  $X_i$  be the RV associated with step  $i$  of the algorithm and defined as follows:

$$X_i = \begin{cases} 1 & \text{if at step } i, \text{ we picked } j \text{ such as } B[j] = \text{null (in one try)} \\ 0 & \text{otherwise.} \end{cases}$$

We have  $\mathbb{P}[X_i = 1] = \frac{n-i+1}{n}$

Now let  $Y_i$  be the number of tries we need at each step  $i$ , to find a null element in  $B$ . Obviously, the running time  $T$  could be expressed as follows:  $T = \sum_{i=1}^n Y_i$ , which yields,  $\mathbb{E}[T] = \sum_{i=1}^n \mathbb{E}[Y_i]$

$Y_i$  is a geometric RV, with parameter  $p = \mathbb{P}[X_i = 1]$ , hence,  $\mathbb{E}[Y_i] = \frac{n}{n-i+1}$ , which yields:

$$\begin{aligned} \mathbb{E}[T] &= \sum_{i=1}^n \frac{n}{n-i+1} \\ &= n(1 + \frac{1}{2} + \dots + \frac{1}{n}) \\ &= nH_n \\ &= \Theta(n \log(n)) \end{aligned}$$

**Exercise 5.**

The case where  $n = 1$ , is trivial, we will start our induction from  $n = 2$ .

Let's say  $A = [a_1, a_2]$ , we have two permutations in total,  $A_1 = A$  and  $A_2 = [a_2, a_1]$ . In this case, the main loop of the algorithm has two runs, but the second run (and generally when  $i = n$ ) doesn't not change the outcome of the list. Hence, the list is only changed in the first step ( $i = 1$ ). Which yields two possibilities with equal probability, we either pick  $j = 2$  (the algorithm swipes the two elements, and outputs,  $A_2$ ), or we pick  $j = 1$ , and nothing changes to the list, and the algorithm outputs  $A_1$ .

Now, we suppose that the algorithm is correct for some  $n \geq 2$ , and prove that the algorithm is correct for an input of size  $n + 1$ :

Let,  $T$  be a list of size  $n + 1$ , and let  $L = [L_1, L']$  be a random permutation of  $T$ , where  $L_1$  is the first element of  $L$ , and  $L'$ , is random permutaion of  $T/\{L_1\}$  (i.e., the list  $T$  minus the element  $L_1$ ). We need to prove that  $\text{FastRandomPermutation}(T)$ , has a probability of  $\frac{1}{(n+1)!}$  of outputing  $L$ .

For the first step, the algorithm picks at uniformly random an element of  $T$ , and swipes it with the first elemenet of  $T$  ( $T_1$ ). Hence, we have get a probability of  $\frac{1}{n+1}$  of actually picking  $L_1$ . For the rest, we are left with a list of size  $n$ ,  $T/\{L_1\}$ , and since  $L'$  is a valid permutation of  $T/\{L_1\}$ , we get by induction, that the probability pof picking  $L'$  in the steps  $i \in \{2, \dots, n + 1\}$  is  $\frac{1}{n!}$ .

By the the multiplication theorem of probability, we conclude that the probability of picking  $L$  as a permutaion is  $\frac{1}{n+1} \frac{1}{n!} = \frac{1}{(n+1)!}$ . This completes our proof by induction.