Randomised Algorithms
Winter term 2022/2023, Exercise Sheet No. 7

Authors:
Ben Ayad, Mohamed Ayoub
Kamzon, Noureddine

December 1, 2022

Exercise 1.

Exercise 2.

Exercise 3.

Let X_i be the RV associated with step i of the algorithm and defined as follows:
$X_i = \begin{cases} 1 & \textit{if at step } i\textit{, we picked } j \textit{ such as } B[j] = null \textit{ (in one try)} \\ 0 & \textit{otherwise.} \end{cases}$
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. Abviously, 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:*

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

Exercise 4.

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* $FastRandomPermutation(T)$*, has a probability of* $\frac{1}{(n+1)!}$ *of outputing* L*.*

For the first step, the algorithm picks at uniformely 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* $i \in \{2, \ldots, n+1\}$ *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.*