Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Institute of Theoretical Computer Science
Mohsen Ghaffari, Angelika Steger, Emo Welzl, Peter Widmayer

Algorithms, Probability, and Computing **Solutions KW40** **HS16**

Solution 1: Number of Leaves

Let L_n be the number of leaves and $l_n := \mathbf{E}[L_n]$.

Clearly, $l_0 = 0$ and $l_1 = 1$. Now for larger n, if the root has rank k, then the number of leaves is the sum of the number of leaves in the left subtree and the number of leaves in the right subtree. We thus get

$$\mathbf{E}[L_n] = \sum_{k=1}^{n} \underbrace{\mathbf{E}[L_n \mid \text{rk(root)} = k]}_{l_{k-1}+l_{n-k}} \cdot \underbrace{\Pr[\text{rk(root)} = k]}_{\frac{1}{n}}.$$

Therefore,

$$l_n = \begin{cases} 0 & \text{if } n = 0, \\ 1 & \text{if } n = 1, \\ \frac{2}{n}\sum_{k=0}^{n-1} l_k & \text{if } n \geq 2. \end{cases}$$

To solve the recurrence, first we compute that $l_2 = 1$. Now for $n \geq 3$, we multiply the recurrence relation by n and write it once for n and once for $n-1$. This yields

$$nl_n = 2\sum_{i=1}^{n-1} l_i \tag{1}$$

and

$$(n-1)l_{n-1} = 2\sum_{i=1}^{n-2} l_i. \tag{2}$$

Now subtracting (2) from (1), we obtain

$$nl_n - (n-1)l_{n-1} = 2l_{n-1}$$

and thus

$$nl_n = (n+1)l_{n-1}.$$

Dividing by $n(n+1)$ yields

$$\frac{l_n}{n+1} = \frac{l_{n-1}}{n}.$$

Repeated application of this equality demonstrates that

$$\frac{l_n}{n+1} = \frac{l_2}{3} = \frac{1}{3}.$$

Therefore, $l_n = \frac{n+1}{3}$.

Solution 2: Random Decline

(1) 1^{st} **variant**: Let N_n be the random variable for the number of numbers chosen if we start with the first number sampled u.a.r. from $\{1..n\}$. We introduce also $N_0 := 0$ for convenience. Then $\mathbf{E}[N_0] = 0$, and for $n \in \mathbf{N}$

$$\underbrace{\mathbf{E}[N_n]}_{=:a_n} = \sum_{k=1}^{n} \underbrace{\mathbf{E}[N_n \,|\, \text{first number is } k]}_{1+\mathbf{E}[N_{k-1}]} \cdot \underbrace{\Pr[\text{first number is } k]}_{=1/n}$$

That is,

$$a_n = \begin{cases} 0 & \text{if } n = 0; \\ 1 + \frac{1}{n}\sum_{k=1}^{n} a_{k-1} & \text{otherwise.} \end{cases}$$

$\mathbf{E}[N_n] = a_n = H_n$ for all $n \in \mathbf{N}_0$ follows along familiar lines (compare Exercise 1.14 (1)).

2^{nd} **variant**: Instead of going the straightforward way of setting up the recurrence and only in the end noticing that it closely resembles something we have seen before, we could have as well observed the connection to randomized search trees in the first place.

Namely, we claim that the distribution of the random set $\mathcal{K}_n := \{k_1, k_2, \ldots, k_{N_n}\}$ resulting from the random process above is the same as the distribution of the ranks of the keys in the left spine of a randomized search tree T_n on n nodes. Recall that the *left spine* is defined to be the set of nodes on the path from the root to the smallest key in T_n.

To prove the claim, proceed by induction on n. For $n = 1$, $\mathcal{K}_1 = \{1\}$ always. So much for the base case. Now once we have verified for $n > 1$ that for all $n' < n$, $\mathcal{K}_{n'}$ is distributed as the ranks of the keys in the left spine of $T_{n'}$, consider $\mathcal{K}_n$. The first number, k_1, is distributed uniformly among $\{1..n\}$ as is the rank of the root of T_n, which is the first node in the left spine of T_n. Now fix a value of k_1 and condition on it. Recursively, the remainder $\mathcal{K}_n \backslash \{k_1\}$ is distributed like $\mathcal{K}_{k_1-1}$. By induction, $\mathcal{K}_{k_1-1}$ is distributed like the ranks of the keys in the left spine of T_{k_1-1}, which is in turn distributed exactly like the left subtree of T_n conditioned on the root having rank k_1, completing the induction.

With this in mind, the size N_n of the set $\mathcal{K}_n$ has to equal the depth of the smallest key in the corresponding T_n, plus one since the depth does not account for the root. Therefore,

$$\mathbf{E}[N_n] = \mathbf{E}\left[D_n^{(1)}\right] + 1 = (H_n - 1) + 1 = H_n.$$

3^{rd} **variant**: There is also a solution using indicator variables. For $1 \le i \le n$, define

$$B_n^{(i)} = \begin{cases} 1 & \text{if } i \in \mathcal{K}_n \\ 0 & \text{if } i \notin \mathcal{K}_n \end{cases}$$

and show that $\Pr\left[B_n^{(i)} = 1\right] = 1/i$ by induction as follows. The base case for $n = 1$ is trivial. Suppose we have proved for all $n' < n$ and all $1 \le i \le n'$ that $\Pr\left[B_{n'}^{(i)} = 1\right] = 1/i$ and now consider the case of $B_n^{(i)}$ for some i. Conditioning on $k_1 \in \{1..i\}$, the only way for $i \in \mathcal{K}_n$ to occur is that $k_1 = i$, which happens with probability $1/i$. On the other hand, if $k_1 \in \{i+1..n\}$, let $n' = k_1 - 1 < n$ and then $\Pr\left[B_n^{(i)} = 1\right] = \Pr\left[B_{n'}^{(i)}\right] = 1/i$ by virtue of the induction hypothesis. Since the probability is $1/i$ conditioning on any of the two cases, it is $1/i$ globally. Note that, after making the observation from the 2nd VARIANT, $B_n^{(i)}$ is apparently distributed like $A_i^!$ for a randomized search tree on n nodes, so we could have as well saved ourselves the pain of this calculation and looked it up on page 18 of the script. We conclude that

$$\mathbf{E}[N_n] = \mathbf{E}\left[\sum_{i=1}^{n} B_n^{(i)}\right] = \sum_{i=1}^{n} \mathbf{E}\left[B_n^{(i)}\right] = \sum_{i=1}^{n} \Pr\left[B_n^{(i)} = 1\right] = \sum_{i=1}^{n} \frac{1}{i} = H_n,$$

as expected.

(2) 1st **variant**: For $n \in \mathbf{N}$, let $s_n := \mathbf{E}[k_1 + k_2 + \ldots + k_{N_n}]$ be the sum we are after. Define, for convenience, $s_0 := 0$. Conditioning on $k_1 = i$ for some $1 \le i \le n$, we obviously have $s_n = i + s_{i-1}$. Therefore

$$s_n = \sum_{i=1}^{n} (i + s_{i-1}) \frac{1}{n} = \frac{1}{n}\binom{n+1}{2} + \frac{1}{n}\sum_{i=1}^{n} s_{i-1} = \frac{n+1}{2} + \frac{1}{n}\sum_{i=1}^{n} s_{i-1}$$

for $n \ge 1$. The usual procedures (multiplication by n and subtracting the recursion identity for $n-1$ instead of n) leads to $s_n = 1 + s_{n-1}$ for $n \ge 2$. Since $s_1 = 1$, this yields $s_n = n$ for all $n \in \mathbf{N}_0$.

2nd **variant**: An alternative way follows the 3rd VARIANT for (i) where we have proved that $\mathbf{E}\left[B_n^{(i)}\right] = 1/i$. Clearly, we can obtain s_n as

$$s_n = \mathbf{E}\left[\sum_{i=1}^{N_n} k_i\right] = \mathbf{E}\left[\sum_{i=1}^{n} B_n^{(i)} i\right] = \sum_{i=1}^{n} \mathbf{E}\left[B_n^{(i)}\right] i = \sum_{i=1}^{n} \frac{1}{i} i = n.$$

Solution 3: Maximum Expectation vs. Expected Maximum

(a) If we are allowed to have the variables depend on one another, this task is not too difficult. Just make sure that there is always one random variable taking value n. For example, consider a probability space with a random variable I which takes a value from $\{1..n\}$ uniformly at random. Then define the $\{X_i\}_{i \in \{1..n\}}$ as follows:

$$X_i := \begin{cases} n & \text{if } I = i, \\ 0 & \text{else.} \end{cases}$$

This way, obviously each X_i takes value 0 with probability $1 - 1/n$ and value n with probability $1/n$ which makes for an expectation of $\mathbf{E}[X_i] = 1$, independent of n. On the other hand, we clearly have $\max_i X_i = n$ with probability 1 and therefor $\mathbf{E}[\max_i X_i] = n$ as required.

(b) If we are not allowed to introduce dependencies among the variables, the task gets slightly more difficult. We claim that the dependencies in example (a) were not really necessary and that the following definition of $\{X_i\}_{i\in\{1..n\}}$ serves the purpose:

$$X_i := \begin{cases} 2n & \text{with probability } 1/n \\ 0 & \text{with probability } 1 - 1/n, \end{cases}$$

where the X_i are i.i.d. (independent and identically distributed). $\mathbf{E}[X_i]$ is just constantly larger than in (a): we have an expectation of exactly 2. Now however, we are left with the task of estimating $\mathbf{E}[\max_i X_i]$. Clearly, we have

$$\max_i X_i = \begin{cases} 2n & \text{if } \exists i : X_i = 2n \\ 0 & \text{else.} \end{cases}$$

Thus $\mathbf{E}[\max_i X_i] = 2n \cdot \Pr[\exists i : X_i = 2n]$. We must now calculate this probability. To this end, we observe that

$$\Pr[\exists i : X_i = 2n] = 1 - \Pr[\forall i : X_i = 0] = 1 - \left(1 - \frac{1}{n}\right)^n. \tag{3}$$

If we can prove that this amount is sufficiently bounded from below by then we are done. Indeed, from the well-known inequality $\forall x : 1 + x \leq e^x$ we get that

$$1 - \frac{1}{n} \leq e^{-1/n}$$

and therefore

$$\left(1 - \frac{1}{n}\right)^n \leq e^{-1}. \tag{4}$$

Combining (3) and (4) yields

$$\Pr[\exists i : X_i = 2n] \geq 1 - e^{-1}$$

and therefore $\mathbf{E}[\max_i X_i] \geq (1 - e^{-1})2n > n$, as required.

Solution 4: Size of Subtrees

(1) Clearly this follows from (2), so a valid way to proceed is to solve (2) first. But there is in fact a stronger relation between the two quantities. In fact, we claim that

$$\sum_{i=1}^{n} W_n^{(i)} = n + \sum_{i=1}^{n} D_n^{(i)}$$

so these are the exact same random variables, i.e. they map every binary search tree to the same number.

There are several ways to see this. One of them is the counting argument described in the script, where each node has an account, node i starting with a balance of $D_n^{(i)} + 1$, then each node 'travels' from its position along the path to the root and leaves one coin at every node it visits, resulting in every node having a balance of $W_n^{(i)}$ in the end (for the details, please see Section 1.5 in the script).

Another way to prove it is by using the usual indicator variables A_i^j. We clearly have

$$\begin{aligned} W_n^{(i)} &= \sum_{j=1}^{n} A_j^i \\ D_n^{(i)} &= \left(\sum_{j=1}^{n} A_i^j \right) - 1. \end{aligned}$$

Therefore

$$\begin{aligned} \sum_{i=1}^{n} W_n^{(i)} &= \sum_{i=1}^{n} \sum_{j=1}^{n} A_j^i \\ \sum_{i=1}^{n} D_n^{(i)} &= \sum_{i=1}^{n} \left(\left(\sum_{j=1}^{n} A_i^j \right) - 1 \right) = \left(\sum_{i=1}^{n} \sum_{j=1}^{n} A_i^j \right) - n, \end{aligned}$$

from which the desired equality follows by inverting the order of summation (even without knowledge of the distribution of A_i^j).

(2) We observe $W_n^{(i)} = \sum_{j=1}^{n} A_j^i$. We employ $\mathbf{E}\left[A_j^i\right] = \mathbf{E}\left[A_i^j\right]$ (cf. Lemma 1.5 in the script) and so

$$\mathbf{E}\left[W_n^{(i)}\right] = \sum_{j=1}^{n} \mathbf{E}\left[A_j^i\right] = \sum_{j=1}^{n} \mathbf{E}\left[A_i^j\right] = 1 + \mathbf{E}\left[D_n^{(i)}\right] .$$

Thus, these variables, too, happen to have the same expectation. Note that contrary to what we observed for (1), $W_n^{(i)}$ and $1 + D_n^{(i)}$ are *not* the same random variables, they are not even identically *distributed*. That they are not equal is trivial. To find a mismatch in the distributions, for instance consider $\Pr\left[W_n^{(1)} = n\right]$ and $\Pr\left[1 + D_n^{(1)} = n\right]$. For the subtree size, $W_n^{(1)} = n$ iff the smallest key becomes the root, therefore

$$\Pr\left[W_n^{(1)} = n\right] = \Pr\left[\text{rk(root)} = 1\right] = \frac{1}{n}.$$

For the depth, the smallest key can have depth $n - 1$ only in one specific case, namely when the left spine of the tree contains all available nodes, thus when the tree is a path, each node being the left child of its parent. By Lemma 1.1, the probability of such a tree is $1/n!$, hence

$$\Pr\left[1 + D_n^{(1)} = n\right] = \Pr\left[\text{left spine has } n \text{ nodes}\right] = 1/n!$$

and thus the two distributions cannot be identical.

(3) For all i, we have $1 \leq W_n^{(i)} \leq n$. There has to be one node that is the root, which has n nodes in its subtree. Consequently, $\max\{W_n^{(i)} | i \in \{1..n\}\} = n$, always, and so

$$\mathbf{E}\left[\max_{i=1}^{n} W_n^{(i)}\right] = n.$$

Again, we can look at the related expression for the depth. There, we know from Section 1.3 in the script that

$$\mathbf{E}\left[\max_{i=1}^{n}\left(1 + D_n^{(i)}\right)\right] \leq 1 + 4.312 \ln n,$$

yielding another proof for the two distributions to be unequal. Please note as well that the max-operator does not commute with the expectation, as for instance in the present example,

$$\max_{i=1}^{n} \mathbf{E}\left[W_n^{(i)}\right] = \max_{i=1}^{n} \mathbf{E}\left[1 + D_n^{(i)}\right].$$

So we see that expectations of maxima can differ quite significantly from maxima of expectations.

Solution 5: Advanced Recurrences

(a) We use the transformation $b_n := \log a_n$ for all n. Note that $\forall n : a_n \geq 1$ is evident from the recurrence relation, therefore the transformation is bijective and does not introduce any spurious solutions. Now we have

$$b_n = \begin{cases} 1 & \text{if } n = 1, \\ 2 + \sum_{j=1}^{n-1} b_j & \text{if } n \geq 2. \end{cases}$$

We note that this is the recurrence from Exercise 5 (2) from KW39. The solution there was

$$b_n = \begin{cases} 1 & \text{if } n = 1, \\ 3 \cdot 2^{n-2} & \text{if } n \geq 2. \end{cases}$$

Therefore

$$a_n = \begin{cases} 2 & \text{if } n = 1, \\ 8^{2^{n-2}} & \text{if } n \geq 2. \end{cases}$$

(b) Subtracting the recurrence for n and $n-1$ (for $n \geq 2$), we obtain

$$b_n - b_{n-1} = 2(-1)^n b_{n-1},$$

or equivalently

$$b_n = (2(-1)^n + 1)b_{n-1}$$

for $n \geq 2$. Expanding the multiplication, we get

$$b_n = \begin{cases} 3^{n/2}(-1)^{n/2-1}b_1 & \text{for } n \text{ even,} \\ (-3)^{(n-1)/2}b_1 & \text{for } n \text{ odd.} \end{cases}$$

Together with $b_1 = 1 + 2(-1)7 = -13$, this yields the final result

$$b_n = \begin{cases} 13(-3)^{n/2} & \text{for } n \text{ even,} \\ -13(-3)^{(n-1)/2} & \text{for } n \text{ odd.} \end{cases}$$

for $n \geq 2$.

(c) As the hint suggests, in such a case, we proceed by coming up with a conjecture and demonstrating its correctness via induction. Let us have a look at the first few numbers of the sequence:

$$1, 3, 3, 7, 7, 15, 15, 31, 31, ...$$

We notice that in every second step, the number doubles and increases additionally by one whilst every other step it does not change. In writing, we conjecture that

$$c_n = \begin{cases} 2^{n/2+1} - 1 & \text{if } n \text{ is even} \\ 2^{(n+1)/2+1} - 1 & \text{if } n \text{ is odd.} \end{cases}$$

To be sure that we are correct, an induction is now needed. The base cases for $n = 0, 1, 2$ are easily verified. Then for $n \geq 3$, if the induction hypothesis holds, then if n is even,

$$c_n = 2^{n/2+1} - 1 + 2 \cdot 2^{(n-2)/2+1} - 2 - 2 \cdot 2^{(n-2)/2+1} + 2 = 2^{n/2+1} - 1$$

and if it is odd

$$c_n = 2^{(n-1)/2+1} - 1 + 2 \cdot 2^{(n-1)/2+1} - 2 - 2 \cdot 2^{(n-3)/2+1} + 2 = 2^{(n+1)/2+1} - 1,$$

as claimed. So far for the mandatory part.

Remark. Sometimes it is not so easy to find the right conjecture, after all, the recursion might bring forth a much more intricate pattern. Let us sketch how it is possible to come up with a good conjecture in a way that is rather generic (works for other recursions of a similar type too).

If a recursion is linear as in our example (i.e. of the form $c_n = a_1 c_{n-1} + a_2 c_{n-2} + ... + a_k c_{n-k}$ with constant coefficients a_j), then we expect the solution to be a superposition of exponential functions and make the ansatz $c_j = x^j$. The recurrence then yields

$$x^n = x^{n-1} + 2x^{n-2} - 2x^{n-3}.$$

Dividing by x^{n-3}, this yields

$$x^3 = x^2 + 2x - 2,$$

or, in normal form, the cubic equation

$$x^3 - x^2 - 2x + 2 = 0.$$

We figure without much pain that $x = 1$ satisfies the equation, so factoring out $(x-1)$ yields

$$(x-1)(x^2-2) = 0.$$

The other two solutions are thus $x = \pm\sqrt{2}$. These three solutions being candidates, we conjecture that the solution of the recurrence will have the form

$$c_n = A \cdot 1^n + B\sqrt{2}^n + C(-\sqrt{2})^n$$

for suitable coefficients A, B and C. Since we have three initial conditions to satisfy, we can produce the three equations

$$\begin{aligned} A \cdot 1^0 + B\sqrt{2}^0 + C\sqrt{2}^0 &= 1 \\ A \cdot 1^1 + B\sqrt{2}^1 - C\sqrt{2}^1 &= 3 \\ A \cdot 1^2 + B\sqrt{2}^2 + C\sqrt{2}^2 &= 3 \end{aligned}$$

from which we conclude that $A = -1$, $B = 1 + \sqrt{2}$ and $C = 1 - \sqrt{2}$. We therefore conjecture that the solution of the recurrence be

$$c_n = (1+\sqrt{2})\sqrt{2}^n + (1-\sqrt{2})(-\sqrt{2})^n - 1.$$

This can be simplified by noting that depending on the parity of n, many terms cancel out. If we make a case distinction on whether n be even or odd, we arrive at the conjecture we made above.