Algorithms, Probability and Computing: Special Assignment 1

Due 25 October 2015

Kevin Klein

Exercise 1

For this exercise, we assume that the relevant random binary search tree holds the set of elements S of size n. Without loss of generality, we will assume that $S = \{1, n\}$ and in particular $i \in S \Rightarrow rank(i) = i$. Let us the following recursive notation to express the structure of a binary search tree: $(node)(leftSubtree)(rightSubtree)$.

a) By direct application of definitions:
$\mathbb{E}[Z_2^{(1)}] = \frac{1}{2} \cdot \mathbb{E}[Z_2^{(1)}|root = 1] + \frac{1}{2} \cdot \mathbb{E}[Z_2^{(1)}|root = 2] = \frac{1}{2} + \frac{1}{2} = 1$
$\mathbb{E}[Z_2] = \mathbb{E}[\frac{1}{2-1} \cdot \sum_{i=1}^{2-1}(Z_2^{(i)})] = \mathbb{E}[Z_2^1] = 1$

$$\mathbb{E}[Z_3^{(1)}] = \frac{1}{6} \cdot \mathbb{E}[Z_3^{(1)}|(1)(2)(3)] + \frac{1}{6} \cdot \mathbb{E}[Z_3^{(1)}|(1)(3)(2)] + \frac{1}{3} \cdot \mathbb{E}[Z_3^{(1)}|(2)(1)(3)] \tag{1}$$
$$+ \frac{1}{6} \cdot \mathbb{E}[Z_3^{(1)}|(3)(1)(2)] + \frac{1}{6} \cdot \mathbb{E}[Z_3^{(1)}|(3)(2)(1)] \tag{2}$$
$$= \frac{1}{6} \cdot (1 + 2 + 2 + 1 + 1) = \frac{7}{6} \tag{3}$$

$$\mathbb{E}[Z_3] = \mathbb{E}[\frac{1}{3-1} \cdot \sum_{i=1}^{3-1} Z_3^{(i)}] = \frac{1}{2} \cdot \mathbb{E}[Z_3^{(1)}] + \frac{1}{2} \cdot \mathbb{E}[Z_3^{(2)}] \tag{4}$$
$$= \frac{1}{2} \cdot (\frac{7}{6} + \frac{1}{6} \cdot \mathbb{E}[Z_3^{(2)}|(1)(2)(3)] + \frac{1}{6} \cdot \mathbb{E}[Z_3^{(2)}|(1)(3)(2)] + \frac{1}{3} \cdot \mathbb{E}[Z_3^{(2)}|(2)(1)(3)] \tag{5}$$
$$+ \frac{1}{6} \cdot \mathbb{E}[Z_3^{(2)}|(3)(1)(2)] + \frac{1}{6} \cdot \mathbb{E}[Z_3^{(1)}|(3)(2)(1)]) \tag{6}$$
$$= \frac{1}{2} \cdot (\frac{7}{6} + \frac{7}{6}) \tag{7}$$
$$= \frac{7}{6} \tag{8}$$

b) We learnt from a) that $\mathbb{E}[Z_2^{(1)}] = 1$ and $\mathbb{E}[Z_3^{(1)}] = \frac{7}{6}$.
For $n \geq 4$:

$$\mathbb{E}[Z_n^{(1)}] = \sum_{i=1}^{n}(\Pr(root = i) \cdot \mathbb{E}[Z_n^{(1)}|root = i]) \tag{9}$$
$$= \frac{1}{n} \cdot (\sum_{i=1}^{n} \mathbb{E}[Z_n^{(1)}|root = i])) \tag{10}$$
$$= \frac{1}{n} \cdot (\mathbb{E}[Z_n^{(1)}|root = 1] + \mathbb{E}[Z_n^{(1)}|root = 2] + \sum_{i=3}^{n} \mathbb{E}[Z_n^{(1)}|root = i])) \tag{11}$$

If 1 is the root, we know that 2 is in the right subtree of the root. Furthermore we can observe that the distance between 1 and 2 corresponds to the depth of 2 in the right subtree incremented by one. Note that 2 has the rank 1 in the right subtree because 1 sits in the root. If 2 is the root, we know that 1 is in its left subtree. Furthermore we know that $S = 1, n$, which implies that $S \cap (1, 2) = \emptyset$. Hence 1 is the only node in the left subtree of 2. Therefore there distance is equal to 1. In all other cases, 1 and 2 are in the left subtree of the root and there distance is equal to their distance in the left subtree of the root as the root cannot be on the path between 1 and 2 as it is larger than both.

Moreover we learned that $D_n^{(i)} = H_i - H_{n-i+1} - 2$ [0].

$$\mathbb{E}[Z_n^{(1)}] = \frac{1}{n} \cdot (D_{n-1}^{(1)} + 1 + 1 + \sum_{i=3}^{n} \mathbb{E}[Z_{i-1}^{(1)}]) \tag{12}$$

$$= \frac{1}{n} \cdot (H_1 + H_{n-1-1+1} + \sum_{i=3}^{n} \mathbb{E}[Z_{i-1}^{(1)}]) \tag{13}$$

$$= \frac{1}{n} \cdot (H_1 + H_{n-1} + \sum_{i=3}^{n} \mathbb{E}[Z_{i-1}^{(1)}]) \tag{14}$$

$$n \cdot \mathbb{E}[Z_n^{(1)}] - (n-1) \cdot \mathbb{E}[Z_{n-1}^{(1)}] = H_{n-1} - H_{n-2} + \mathbb{E}[Z_{n-1}^{(1)}] \tag{15}$$

$$\Leftrightarrow n \cdot \mathbb{E}[Z_n^{(1)}] = n \cdot \mathbb{E}[Z_{n-1}^{(1)}] + \frac{1}{n-1} \tag{16}$$

$$\Leftrightarrow \mathbb{E}[Z_n^{(1)}] = \mathbb{E}[Z_{n-1}^{(1)}] + \frac{1}{n(n-1)} \tag{17}$$

$$= \frac{1}{n(n-1)} + \frac{1}{(n-1)(n-2)} + ... + \frac{1}{4 \cdot 3} + \mathbb{E}[Z_3^{(1)}] \tag{18}$$

$$= \sum_{i=4}^{n} \frac{1}{i(i-1)} + \mathbb{E}[Z_3^{(1)}] \tag{19}$$

$$= \sum_{i=4}^{n} \frac{i-(i-1)}{i(i-1)} + \mathbb{E}[Z_3^{(1)}] \tag{20}$$

$$= \sum_{i=4}^{n} (\frac{1}{i-1} - \frac{1}{i}) + \mathbb{E}[Z_3^{(1)}] \tag{21}$$

$$= \sum_{i=4}^{n} \frac{1}{i-1} - \sum_{i=4}^{n} \frac{1}{i} + \mathbb{E}[Z_3^{(1)}] \tag{22}$$

$$= \sum_{i=3}^{n-1} \frac{1}{i} - \sum_{i=4}^{n} \frac{1}{i} + \mathbb{E}[Z_3^{(1)}] \tag{23}$$

$$= \frac{1}{3} - \frac{1}{n} + \mathbb{E}[Z_3^{(1)}] \tag{24}$$

$$= \frac{1}{3} - \frac{1}{n} + \frac{7}{6} \tag{25}$$

$$= \frac{3}{2} - \frac{1}{n} \tag{26}$$

We see that $\frac{3}{2} - \frac{1}{2} = 1$ and $\frac{3}{2} - \frac{1}{3} = \frac{7}{6}$, therefore we can conclude that

$$\forall n \geq 2 : \mathbb{E}[Z_n^{(1)}] = \frac{3}{2} - \frac{1}{n}$$

c)

$$Z_n^{(k)} = |D_n^{(k)} - D_n^{(k+1)}| \text{ (definition of } Z_n^k) \tag{27}$$

$$= |\sum_{i=1}^{n} A_i^{(k)} - \sum_{i=1}^{n} A_i^{(k+1)}| \text{ (definition of } D_n^{(k)}) \tag{28}$$

$$= |\sum_{i=1}^{n} (A_i^{(k)} - A_i^{(k+1)})| \tag{29}$$

$$\tag{30}$$

As we know that k and $k+1$ are ancestors of each other, the terms of the sum are either all ≤ 0 or ≥ 0. Thererfore we can simplify:

$$Z_n^{(k)} = \sum_{i=1}^{n} |A_i^{(k)} - A_i^{(k+1)}| \tag{31}$$

$$\mathbb{E}[Z_n^{(k)}] = \sum_{i=1}^{n} |\mathbb{E}[A_i^{(k)}] - \mathbb{E}[A_i^{(k+1)}]| \tag{32}$$

$$= \sum_{i=1}^{n} |\frac{1}{|i-k|+1} - \frac{1}{|i-k-1|+1}| \text{ (as we have proven in [1])} \tag{33}$$

$$= \sum_{i=1}^{k} (\frac{1}{k-i+1} - \frac{1}{k+1-i+1}) + \sum_{i=k+1}^{n} (\frac{1}{i-k+1} - \frac{1}{i-k-1+1}) \tag{34}$$

$$= \sum_{i=1}^{k} (\frac{1}{k-i+1} - \frac{1}{k-i+2}) + \sum_{i=k+1}^{n} (\frac{1}{i-k+1} - \frac{1}{i-k}) \tag{35}$$

$$= \sum_{i=0}^{k-1} \frac{1}{k-i} - \sum_{i=-1}^{k-2} \frac{1}{k-i} + \sum_{i=1}^{n-k} \frac{1}{i} - \sum_{i=2}^{n-k+1} \frac{1}{i} \tag{36}$$

$$= 1 - \frac{1}{k+1} + 1 - \frac{1}{n-k+1} \tag{37}$$

$$= 2 - \frac{1}{k+1} - \frac{1}{n-k+1} \tag{38}$$

We plug this into the expected value of Z_n.

$$\mathbb{E}[Z_n] = \frac{1}{n-1} \sum_{k=1}^{n-1} \mathbb{E}[Z_n^{(k)}] \tag{39}$$

$$= \frac{1}{n-1} \sum_{k=1}^{n-1} (2 - \frac{1}{k+1} - \frac{1}{n-k+1}) \tag{40}$$

$$= \frac{1}{n-1} (2 - \sum_{k=1}^{n-1} \frac{1}{k+1} - \sum_{k=1}^{n-1} \frac{1}{n-k+1}) \tag{41}$$

$$= \frac{1}{n-1} (2 - \sum_{k=2}^{n} \frac{1}{k} - \sum_{k=2}^{n} \frac{1}{k}) \tag{42}$$

$$= \frac{1}{n-1} (2 - H_n + 1 - H_n + 1) \tag{43}$$

$$= \frac{2}{n-1} (2 - H_n) \tag{44}$$

$$\tag{45}$$

d) We are looking for the probability with which at least one of the three trees has $[Z_n] > 3$. Let's denote

$\Pr[tree\ of\ size\ n\ has\ [Z_n] > 3]$ as p.

$$\mathbb{E}[number\ of\ trees\ with [Z_n] > 3] = 0 \cdot p^3 + 1\binom{3}{1}(p)^2(1-p) + 2\binom{3}{2}(p)(1-p)^2 + 3(1-p)^3 \tag{46}$$
$$= 3p^2 - 3p^3 + 6p(1 - 2p + p^2) + 3(1 - 3p + 3p^2 - p^3) \tag{47}$$
$$= 3p^2 - 3p^3 + 6p - 12p^2 + 6p^3 = 3 - 9p + 9p^2 - 3p^3 \tag{48}$$
$$= 3(1-p) \tag{49}$$
$$\mathbb{E}[number\ of\ trees\ with [Z_n] > 3] \geq 1 \Leftrightarrow 3(1-p) \geq 1 \tag{50}$$
$$\Leftrightarrow p \leq \frac{2}{3} \tag{51}$$
$$\tag{52}$$

In order to show that this holds, let's imply a contradiction by assuming $p > \frac{2}{3}$.
Recall that the expected value of Z_n can be calculated by

$$\mathbb{E}[Z_n] = \sum_{t\ tree\ of\ size\ n} (Z_n\ of\ t) \cdot \Pr[t] \tag{53}$$
$$= \sum_{t\ tree\ of\ size\ n\ with\ Z_n > 3} (Z_n\ of\ t) \cdot \Pr[t] + \sum_{t\ tree\ of\ size\ n\ with\ Z_n \leq 3} (Z_n\ of\ t) \cdot \Pr[t] \tag{54}$$
$$\geq \sum_{t\ tree\ of\ size\ n\ with\ Z_n > 3} 3 \cdot \Pr[t] + \sum_{t\ tree\ of\ size\ n\ with\ Z_n \leq 3} (Z_n\ of\ t) \cdot \Pr[t] \tag{55}$$
$$\geq 3 \cdot \sum_{t\ tree\ of\ size\ n\ with\ Z_n > 3} \Pr[t] + \sum_{t\ tree\ of\ size\ n\ with\ Z_n \leq 3} (Z_n\ of\ t) \cdot \Pr[t] \tag{56}$$
$$\geq 3 \cdot p + \sum_{t\ tree\ of\ size\ n\ with\ Z_n \leq 3} (Z_n\ of\ t) \cdot \Pr[t] \tag{57}$$
$$\geq 3 \cdot p \tag{58}$$
$$> 3 \cdot \frac{2}{3} \tag{59}$$
$$> 2 \tag{60}$$

This contradicts the knowledge we have about $\mathbb{E}[Z_n]$, namely that it is ≤ 2. It follows that $p > \frac{2}{3}$, which is equivalent to saying that we have at least one tree among those tree which has $Z_n < 3$.

[0] APC script 2015 Edition, Chapter 1.4 Expected Depth of Individual Keys, Theorom 1.6, p.18 [1] APC script 2015 Edition, Chapter 1.4 Expected Depth of Individual Keys, Lemma 1.5, p.18

Exercise 2

a) Analogously to the proof of Lemma 1.5 [0], we will proceed with an exhaustive case distinction in order to prove that

$$\mathbb{E}[A_i^j] = \frac{p(j)}{\sum_{l=min\{i,j\}}^{max\{i,j\}} p(l)}$$

(i) $i = j$

$$\mathbb{E}[A_i^{(j)}] = E[A_j^{(j)}] \tag{61}$$
$$= 1 \text{ (definition of ancestor indicator variable)} \tag{62}$$
$$= \frac{p(j)}{p(j)} \ (p(j) > 0 \text{ is given}) \tag{63}$$
$$= \frac{p(j)}{\sum_{l=min\{i,j\}}^{max\{i,j\}} p(l)} \ (\text{as } i = j) \tag{64}$$
$$\tag{65}$$

(ii) $i = 1$ and $j = n$
Note that i and j will always end up in different subtrees of the root, except for the cases in which either of both is the root. Hence, if none of them is the root, the indicator variable is equal to 0. If i is the root, j is a child of i and j is different from i because $n \geq 2$. Therefore j cannot be an ancestor of i if i is the root. Consequently, the only case is which j is an ancestor of i is when j is the root of the tree.

$$\mathbb{E}[A_i^{(j)}] = \Pr(root = j) \tag{66}$$
$$= \frac{p(j)}{\sum_{l=1}^{n} p(l)} \text{ (definition of the weighted binary tree)} \tag{67}$$
$$= \frac{p(j)}{\sum_{l=min\{i,j\}}^{max\{i,j\}} p(l)} \ (\text{as } i = 1 \text{ and } j = n) \tag{68}$$
$$\tag{69}$$

(iii) $j = 1$ and $i = n$
This case is symmetric to case (ii).

(iv) $i \neq j$ and $n > j - i + 1$ and $root \in \{min\{i,j\}, max\{i,j\}\}$
If i is the root, j cannot be an ancestor of as we know that $i \neq j$. If $root \in \{min\{i,j\}+1, max\{i,j\}-1\}$, we know that i and j will always end up in different subtrees of the root. Hence, if none of them is the root, the indicator variable is equal to 0. Consequently, the only case is which j is an ancestor of i is when j is the root of the tree.

$$\mathbb{E}[A_i^{(j)}] = \Pr(root = j | root \in \{min\{i,j\}, max\{i,j\}\}) \tag{70}$$
$$= \frac{\Pr(root = j)}{\Pr(root \in \{min\{i,j\}, max\{i,j\}\})} \text{ (definition of conditional probability)} \tag{71}$$
$$= \frac{\frac{p(j)}{\sum_{l=1}^{n} p(l)}}{\frac{\sum_{l=min\{i,j\}}^{max\{i,j\}} p(l)}{\sum_{l=1}^{n} p(l)}} \text{ (definition of the weighted binary tree)} \tag{72}$$
$$= \frac{p(j)}{\sum_{l=min\{i,j\}}^{max\{i,j\}} p(l)} \tag{73}$$

(v) $n > j - i + 1$ and $root \notin \{min\{i,j\}, max\{i,j\}\}$
As the root must either be smaller than i and smaller than j or larger than i and larger than j, we know that i and j will always end up in the same subtree of the root. Note that within the left subtree the ranks of the keys will remain the same. Within the right subtree, the ranks are all decreased by k where $k = root$. Hence the difference of ranks between i and j remains the same in both subtrees.
Therefore, we can now look at the respective subtree containing i and j in an isolated fashion, as

we know that the current root does not contribute anything to the notion we look at. In the base case in which $n = 2$, we can either apply (i) if i and j are equal and (ii) if they're not. For $n > 2$, we can inductively apply cases (i), (ii), (iii), (iv) and (v).

b) We want to prove that

$$\forall x, y \in \mathbb{R}\ s.t.\ 0 \leq x < y : \frac{x}{y} \leq ln(y) - ln(y - x)$$

As we know that $x < y$, we can formulate the following

$$\exists\ \delta \in \mathbb{R}\ s.t.\ \delta > 0\ and\ x = y - \delta$$

We know that $\forall x \in \mathbb{R} : 1 + z \leq e^z$. With $z = ln(\frac{\delta}{y})$ we get the following:

$$1 + ln(\frac{\delta}{y}) \leq e^{ln(\frac{\delta}{y})} \tag{74}$$

$$\Leftrightarrow 1 - ln(\frac{y}{\delta}) \leq \frac{\delta}{y} \tag{75}$$

$$\Leftrightarrow 1 - \frac{\delta}{y} \leq ln(\frac{y}{\delta}) \tag{76}$$

$$\Leftrightarrow \frac{y - \delta}{y} \leq ln(\frac{y}{y - y + \delta}) \tag{77}$$

$$\Leftrightarrow \frac{x}{y} \leq ln(\frac{y}{y - x - \delta + \delta}) \tag{78}$$

$$\Leftrightarrow \frac{x}{y} \leq ln(\frac{y}{y - x}) \tag{79}$$

$$\Leftrightarrow \frac{x}{y} \leq ln(y) - ln(y - x) \tag{80}$$

$$\tag{81}$$

c) We want to prove that

$$\forall i \in [n] : \mathbb{E}[D_n^{(i)}] \leq 2 \cdot ln(\frac{1}{p(i)})$$

$$\mathbb{E}[D_n^{(i)}] = \sum_{j=1}^{n} \mathbb{E}[A_i^j] \text{ (definition of Depth } D_n^{(i)}) \tag{82}$$

$$= \sum_{j=1, j\neq i}^{n} \frac{p(j)}{\sum_{l=min\{i,j\}}^{max\{i,j\}} p(l)} \text{ (application of a))} \tag{83}$$

$$\leq \sum_{j=1, j\neq i}^{n} ln(\sum_{l=min\{i,j\}}^{max\{i,j\}} p(l)) - ln(\sum_{l=min\{i,j\}}^{max\{i,j\}} p(l) - p(j)) \text{ (application of b))} \tag{84}$$

$$\leq (\sum_{j=1, j\neq i}^{n} ln(\sum_{l=min\{i,j\}}^{max\{i,j\}} p(l))) - ((\sum_{j=1, j\neq i}^{n} ln(\sum_{l=min\{i,j\}}^{max\{i,j\}} p(l) - p(j))) \tag{85}$$

$$\leq \sum_{j=1}^{i-1} ln(\sum_{l=j}^{i} p(l))) + \sum_{j=i+1}^{n} ln(\sum_{l=i}^{j} p(l))) - \sum_{j=1}^{i-1} ln(\sum_{l=j}^{i} p(l) - p(j))) - \sum_{j=i+1}^{n} ln(\sum_{l=i}^{j} p(l) - p(j))) \tag{86}$$

$$\leq \sum_{j=1}^{i-1} ln(\sum_{l=j}^{i} p(l))) + \sum_{j=i+1}^{n} ln(\sum_{l=i}^{j} p(l))) - \sum_{j=1}^{i-1} ln(\sum_{l=j+1}^{i} p(l))) - \sum_{j=i+1}^{n} ln(\sum_{l=i}^{j-1} p(l))) \tag{87}$$

$$\leq \sum_{j=1}^{i-1} ln(\sum_{l=j}^{i} p(l))) + \sum_{j=i+1}^{n} ln(\sum_{l=i}^{j} p(l))) - \sum_{j=2}^{i} ln(\sum_{l=j}^{i} p(l))) - \sum_{j=i}^{n-1} ln(\sum_{l=i}^{j} p(l))) \tag{88}$$

$$\leq ln(\sum_{l=1}^{i} p(l)) - ln(p(i)) + ln(\sum_{l=i+1}^{n} p(l)) - ln(p(i)) \tag{89}$$

$$\leq ln(\frac{(\sum_{l=1}^{i} p(l))}{p(i)} \cdot \frac{\sum_{l=i+1}^{n} p(l)}{p(i)}) \tag{90}$$

$$\tag{91}$$

The definition of the weighted binary tree tells us that $\sum_{l=1}^{n} p(l) = 1$ as well as $\forall i \in [n], p(i) > 0$. It follows immediately that $\sum_{l=1}^{i} p(l) \leq \sum_{l=1}^{n} p(l) = 1$ and analogously $\sum_{l=i+1}^{n} p(l) \leq \sum_{l=1}^{n} p(l) = 1$. Hence we can conclude:

$$\mathbb{E}[D_n^{(i)}] \leq ln(\frac{1}{p(i)} \cdot \frac{1}{p(i)}) \tag{92}$$

$$\leq 2 \cdot ln(\frac{1}{p(i)}) \tag{93}$$

Exercise 3

a) We determine the maximum number of inversions of permutations, short $maxP(n)$ on a set of n elements by induction.

- Claim: $\forall n \geq 2 : maxP(n) = \frac{n(n-1)}{2}$
- Base case: n = 2:
  There is exactly 1 permutation if the second element is smaller than the first.

$$\frac{n(n-1)}{2} = \frac{2 \cdot 1}{2} = 1$$

- Induction step: $n \to n+1$
  The induction hypothesis tells us that we have $maxP(n) = \frac{n(n-1)}{2}$ for the set of n numbers. We cannot create more than n inversions by adding the $n+1$th element as there are only $n+1$ elements and there

cannot be more than one inversion per pair of elements. We can create exactly n inversions by either positioning an element larger than the previous maximum in the first position or an element being smaller than the previous minimum in last position. Hence

$$\begin{aligned} maxP(n+1) &= maxP(n) + n && (94)\\ &= \frac{n(n-1)}{2} + n \text{ (induction hypothesis)} && (95)\\ &= \frac{n(n-1+2)}{2} && (96)\\ &= \frac{(n+1)n}{2} && (97)\\ & && (98) \end{aligned}$$

b) We are given a set of n non-vertical lines L and $a \in \mathbb{R}$. We assume that the lines are stored in the fashion $l \in L : l = (x, y)$. Note that we consider each intersection of a pair of lines as an intersection. This implies that if k lines intersect in a single point, we count this as $\frac{k(k-1)}{2}$ intersections. We are still not guaranteed to have $\frac{n(n-1)}{2}$ intersections in total, as parallel lines are allowed and a pair of parallel lines does not intersect. An additional relevant observation is that the lines are contained in a set. As a set does not contain duplicates we can conclude that $\forall l_i, l_j \in L, l_i \neq l_j : x_i \neq$ or $y_i \neq y_j$.

1. **Algorithm**
   (1) We convert the set L into a list L, such that from now on, there exists an order of the elements.
   (2) We define a total order in L by defining the permutation $\pi_{-\infty} : L \to \{1..n\}$. This permutation corresponds to relative order of the lines in $x = -\infty$. Intuitively, this order represents the relative order of the lines for $x = -\infty$ in a way that l_i lies above l_j in $x = -\infty$ iff $\pi_{-\infty}(l_i) < \pi_{-\infty}(l_j)$. Formally $\pi_{-\infty}(l_i) < \pi_{-\infty}(l_j) \Leftrightarrow x_i < x_j$ or $(x_i = x_j$ and $y_i > y_j)$. As we do not have duplicates, this order is total and unambiguous.
   (3) We create a permutation $\pi_a : L \to \{1..n\}$ such that it corresponds to relative order of the lines in $x = a$. We arbitrarily define that a line is below the other if it intersects and has previously been above. Formally we sort s.t. $\pi_a(l_i) < \pi_a(l_j) \Leftrightarrow a \cdot x_i + y_i > a \cdot x_j + y_j$ or $(a \cdot x_i + y_i = a \cdot x_j + y_j$ and $x_i > x_j)$.
   (4) We feed the algorithm from Fact A the permutation π_a.
   (5) We return the result from the algorithm of Fact A.
2. **Runtime analysis**
   (1) A constant time operation.
   (2) We first need to evaluate the n lines at a given x, which can be done in $\mathcal{O}(n)$. Sorting n lines can be done by done in $\mathcal{O}(n \cdot log(n))$ time, because we use a constant time comparator.
   (3) Same as previous step.
   (4) The algorithm is given to have $\mathcal{O}(n \cdot log(n))$ runtime.
   (5) Constant time operation.

   As the considered parts of the algorithm are not nested but sequential, the overall runtime is $\mathcal{O}(n \cdot log(n))$.
3. **Correctnes**
   We know that we are given the number of inversions in the permutation π_a, including permutations in $x = a$.
   Hence we need to prove that the amount of inversions in the permutation π_a is equal to the number of intersections of lines from L, up to and including $x = a$. We will do so by induction.

Claim. If there are k lines crossing in one point, the number of inversions in the permutation increases for each pairwise intersection compared to infinitesimally left of this intersection point.
Proof.

- Let's inspect the base case with $k = 2$.
  Given that we have an intersection of lines l_1 and l_2 in $x = a$, we know that $a \cdot x_1 + y_1 = a \cdot x_2 + y_2$. As the permutation is a total order, we know that either $x_1 < x_2$ or vice versa. Let's assume $x_1 < x_2$ (the opposite is symmetric), which implies that $\pi_a(l_2) < \pi_a(l_1)$ according to the definition of π_a.

$$
\begin{aligned}
x_1 < x_2 &\Leftrightarrow x_1 \cdot \delta < x_2 \cdot \delta (\delta \in \mathbb{R}, \delta > 0) && (99)\\
&\Leftrightarrow -x_1 \cdot \delta > -x_2 \cdot \delta && (100)\\
&\Leftrightarrow x_1 \cdot a + y_1 - x_1 \cdot \delta > x_1 \cdot a + y_1 - x_2 \cdot \delta && (101)\\
&\Leftrightarrow x_1 \cdot (a - \delta) + y_1 > x_2 \cdot a + y_2 - x_2 \cdot \delta && (102)\\
&\Leftrightarrow x_1 \cdot (a - \delta) + y_1 > x_2 \cdot (a - \delta) + y_2 && (103)\\
&\Leftrightarrow \pi_{a-\delta}(l_1) < \pi_{a-\delta}(l_2) && (104)\\
& && (105)
\end{aligned}
$$

  Hence we can conclude that there is exactly one inversion of the permutation for one intersection, when considering only two lines. This extends to the general case, namely intersection of precisely two lines in a set of n lines, as we know that the permutation of all other lines is unaffected. We can conclude this by the fact that the permutation represents the above relationship. When we have no further intersections, no further lines can change their above relationship.

- Now let's inspect the case for $k > 2$.
  By the definition of the number of intersections, we have $\binom{k}{2}$ many intersections for k lines in the point with x-value a. Our induction hypothesis tells us that those intersections give us $\binom{k}{2}$ inversions in the permutation. As the new lines intersects all of the lines which already go through the point, we have $\binom{k}{2} + k = \binom{k+1}{2}$ intersections. Given that we have an intersection of lines $l_i with i = 1, .., k$ and $l_k + 1$ in $x = a$, we know that $a \cdot x_i + y_i = a \cdot x_{k+1} + y_{k+1}$. As the permutation is a total order, we know that either $x_i < x_{k+1}$ or vice versa for each individual i. We will simply say that for k_1 many it holds that $x_i < x_{k+1}$ and for $k_2 = k - k_1$ many $x_{k+1} < x_i$. In the analogous way to the base case, we know that

$$
\begin{aligned}
&\forall i \in \{1..k_1\}\ x_i < x_{k+1} && (106)\\
&\Leftrightarrow \forall i \in \{1..k_1\}\ \pi_a(l_{k+1}) < \pi_a(l_i) && (107)\\
&\Leftrightarrow \pi_{a-\delta}(l_i) < \pi_{a-\delta}(l_{k+1}) && (108)\\
& && (109)
\end{aligned}
$$

  Hence we have k_1 inversions by the intersections with the lines $l_i, i \in 1..k_1$. Analogously:

$$
\begin{aligned}
&\forall i \in \{k_2..k\}\ x_{k+1} < x_i && (110)\\
&\Leftrightarrow \forall i \in \{1..k_1\}\ \pi_a(l_i) < \pi_a(l_{k+1}) && (111)\\
&\Leftrightarrow \pi_{a-\delta}(l_{k+1}) < \pi_{a-\delta}(l_i) && (112)\\
& && (113)
\end{aligned}
$$

  Hence we have k_2 inversions by the intersections with the lines $l_i, i \in k_2..n$. As we did not alter the relative order of the k preexisting lines, we now that their permutation has not been influenced by

the addition of the new line l_{k+1}. Thereby we are able to conclude that be are now confronted with $\binom{k}{2} + k = \binom{k+1}{2}$ intersections leading to $\binom{k}{2} + k_1 + k_2 = \binom{k+1}{2}$ inversions in the permutation at the point $x = a$.

Claim. There is no change permutation without intersection.
Proof.
Assume that at point $x = a$ the permutation changes wihtout any lines intersecting in that point. We now that in the previous permutation π_{new} there needs to be at least one additional inversion compared to π_{old}. Say $\pi_{old}(l_i) < \pi_{old}(l_j)(i)$ and $\pi_{new}(l_i) > \pi_{new}(l_j)(i)$. We can simply say that the old permutation was at $x = a - \delta$ where $\delta \in \mathbb{R}, \delta > 0$.
(i) implies that $(a-\delta) \cdot x_i + y_i > (a-\delta) \cdot x_j + y_j$ or $((a-\delta) \cdot x_i + y_i = a \cdot x_j + y_j$ and $x_i > x_j)$. If the latter is true, we know that there is an intersection and the assumption was wrong. (ii) implies that $a \cdot x_j + y_j > a \cdot x_i + y_i$ or $(a \cdot x_j + y_j = a \cdot x_i + y_i$ and $x_j > x_i)$. If the latter is true, we know that there is an intersection and the assumption was wrong. Hence we now assume that $(a-\delta) \cdot x_i + y_i > (a-\delta) \cdot x_j + y_j$ and $a \cdot x_j + y_j > a \cdot x_i + y_i$. We define the function $f : \mathbb{R} \to \mathbb{R} with f(b) = b \cdot x_j + y_j - a \cdot x_i + y_i$. It follows immediately that $f(a-\delta) > 0$ and $f(a) < 0$. As f is steady, the mean value theorem allows us to say that $\exists c \in \mathbb{R} s.t. f(c) = 0$. Therefore $c \cdot x_j + y_j = c \cdot x_i + y_i$, which is equivalent to an intersection. This, again, contradicts our fundamental assumption and proves our claim.

We now know that the permutation only changes when there is an intersection and that its number of inversions is increased by the number of intersections in that intersection point. Hence, if we simply add up all the intersections we have from $x = -\infty$ up to and including $x = a$, this is equivalent to the number of inversions in the permutation from $x = -\infty$, compared to the permutation in $x = a$. As L is ordered in such a fashion as the permutation would be in $= -\infty$, we have proven the correctness of our algorithm. In particular we handle the less general cases of multiple lines intersecting in one point and parallel lines.

c) Let's assume that $b < c$. Our algorithm from b) is able to return us the number of intersections up to a given point $x = a$ by looking at the permutation of the lines in $x = a$, which has been defined in b) as well. By querying the algorithm for $x = a$ and $x = b$ we multiply its runtime by only a constant factor, yet are able to obtain the permutation of the lines in $x = a$ as well as $x = b$.
In order to make use of the algorithm from Fact B, we need to adapt our permutations, as we need to keep in mind that both π_a and π_b can include inversions simultaneously. We are only looking for inversions between both. More precisely, the algorithm will assume that no the baseline of no inversions is $1..n$. To make this fit with the shape of our apply a mapping which maps the ith element of π_b to i to both π_b and π_c. This will make π_c represent the set of inversions from b to c. Therefore we can feed π_c to the algorithm from Fact B and return its result. The exptected runtime remains $\mathcal{O}(nlogn)$ as the algorithm from Fact B takes expected $\mathcal{O}(nlogn)$ runtime after applying our algorithm of $\mathcal{O}(nlogn)$ from b) twice. If $c < b$, we symmetricaly apply the above. If $c = b$, we apply it with $b = c - \delta$ where δ is the smallest number possibly represented and $c = c$.

d) We are looking at the up to $\frac{n(n-1)}{2}$ many lines that pairwise connect two points from our set. Among those lines, we look for the line with the median slope. This means that we are looking for the line $l = (x, y)$ with the median x-value. If we have multiple lines with the same slope, we can just pick any of those.
In order to apply our previous knowledge, we apply the dual transformation of points and lines. In the dual space, our n points are n lines. The lines connecting the n points $p = (x, y)$ correspond to the intersections of the n lines $p^* = (x, -y)$. Therefore, in the dual space, we are looking for one the intersection of lines, which has median x-value.
Algorithm

(1) First we transform all of our n points to the set of lines L by applying the duality transormation.

(2) We don't know whether we have the full $\frac{n(n-1)}{2}$ many intersections, as we allow parallel lines. In order to find out, how many intersections we have in total, we query the algorithm from b) with $x = \infty$. We call the returned result m.

(3) By applying the idea from c) we will do a binary search on intersections until we found the median.
- Set $left$ to $-\infty$ and $right$ to $+\infty$.
- While $left < right$:
  - Query, as in c), a random pair of lines intersecting between $left$ and $right$. Determine their intersection point by solving the linear equation and call the x-value of that point x_I.
  - Query the number of intersections in $-\infty$, up to $x_I - \delta$ and up to x_I by calling the algorithm from b) three separate times with δ the smallest positive, non-zero number.
  - If $result_{x_I} - result_{-\infty} > \frac{m}{2}$ and $result_{x_I - \delta} - result_{-\infty} < \frac{m}{2}$, we know that we have the median intersection in x_I and hence break the while loop. In order to store the result, we set $right = left = x_I$. Else if $result_{x_I} - result_{-\infty} > \frac{m}{2}$ we set $right$ to x_I. Else we set $left$ to $x_I - \delta$.
- When we have $left$ equal to $right$, we know the x of the median intersection point. To get one particular intersection point, we query the algorithm from c) once again with $right$ and $right - \delta$. This will give us any two lines with an intersection point that is exactly on $right$.

(4) Hence we have two lines with an intersection point with median x-value among intersection points in the dual space. In order to return the the points giving the medium slope in the intial space, we transform those lines to points by the duality transformation and return those two points.

Expected runtime

(1) Applying the constant time transformation on n elements takes $\mathcal{O}(n)$ runtime.

(2) One application of the algorithm from b) takes $\mathcal{O}(n \cdot log(n))$ runtime.

(3) Per iteration of the loop we have one application of the algorithm from c) ($\mathcal{O}(n \cdot log(n))$), a constant time solution of a linear equation, and three applications of the algorithm from a) ($\mathcal{O}(n \cdot log(n))$). Hence one loop iteration has expected runtime of $\mathcal{O}(n \cdot log(n))$.
The important question is how many loop iterations we end up having. When looking at the behaviour of the algorithm carefully, we understand that it is performing a randomized binary search on the intersections. The expected number of calls in the binary search on n elements is the equivalent of the depth of a random binary search tree of n elements. We know that the exptected depth of a randomized binary search tree is asymptotically logarithmic in n. As we have up to $\frac{n(n-1)}{2}$ many intersections, we have expected $\mathcal{O}(log(\frac{n(n-1)}{2})) = \mathcal{O}(log(n))$ depth in the binary search tree, respectively as many calls in the binary search.
Followingly, we have expected $\mathcal{O}(log(n))$ loop iterations at expected $\mathcal{O}(n \cdot log(n))$ each, which gives us $\mathcal{O}(n \cdot log^2(n))$

(4) One application of the algorithm from c) takes expected $\mathcal{O}(n \cdot log(n))$ runtime.

As the considered parts of the algorithm are not nested but sequential, the overall expected runtime is $\mathcal{O}(n \cdot log^2(n))$.

Correctness

When the algorithm returns a pair lines, we know that they intersect in $right = left$. $Right$ has either been set to$left$ explicitly or implicitly.
If it has been set explicitly, we found ourselves in the situation where there are $\frac{m}{2}$ many intersections strictly left of $right$ and $\frac{m}{2}$ many from the very left to and including $right$. Therefore we know that the median element has to lie in $right$. As all of the intersections in $right$ are equivalent with respect the median x-coordinate, any of the reported pair of lines intersecting in $right$ is having median $x - value$. Thereby, in

the dual space, the line connecting the two points has median slope.
If it has been set implicitly, we know that either the algorithm has made $left$ pass $right$ or that both are equal. If both are equal, we know that there are no more intersections for which either more than half of the intersections are lying on the left and not more than half of the intersections are lying right. This is the definition of the median. If left has strictly surpassed right, we know the last movement of the right pointer, for some inspected x_R, there were $> \frac{m}{2}$ intersections to the left of x_R. In the last step before $left$ suprassed right, they must have drawn $x_I = right$ as the x_I are drawn from the range $left$ to $right$. As right was not moved in that draw of x_I we know it holds that there are $\leq \frac{m}{2}$ intersections to the left of x_I. Furthermore, as we know that $x_{last} = right$ and that the last x_I $right$ was moved to was x_R it follows $x_{last} = x_R$. As we know that there are $> \frac{m}{2}$ to the left of x_R, this also holds for x_last. this is a contradiction to what we have just established and hence this case cannot happen.