NOTES ON HOOK WALKS

1. The hook length formula

Let f^λ be the number of standard Young tableaux of shape λ. Our first goal will be to prove the *hook length formula* are given by

$$f^\lambda = \frac{|\lambda|!}{\prod_{x\in\lambda} h_\lambda(x)} \tag{1.1}$$

where the product is over all cells x in the Young diagram of λ and $h_\lambda(x)$ is the hook length of the cell x. We'll follow the proof in "A Probabilistic Proof of a Formula for the Number of Young Tableaux of a Given Shape" (1979) by C. Greene, A. Nijenhuis, and H. S. Wilf.

First some examples:

Example 1.1. For the following partitions we write the hook length in each cell and compute f^λ using (1.1).

6	4	3	1
4	2	1	
1			

5	3	2
4	2	1
1		

6	4	2	1
3	1		
1			

5	4	3	1
3	2	1	

$f^{(4,3,1)} = \frac{8!}{6\cdot 4^2\cdot 3\cdot 2} = 70$ $f^{(3,3,1)} = \frac{7!}{5\cdot 4\cdot 3\cdot 2^2} = 21$ $f^{(4,2,1)} = \frac{7!}{6\cdot 4\cdot 3\cdot 2} = 35$ $f^{(5,3)} = \frac{7!}{5\cdot 4\cdot 3^2\cdot 2} = 14$

Note that

$$f^{(4,3,1)} = f^{(3,3,1)} + f^{(4,2,1)} + f^{(5,3)}.$$

The identity in above is an example of a more general recursion relation for the f^λ. To describe this, we will need some definitions. A cell $x \in \lambda$ is a *corner* is there are no cells directly to its right or directly below. Let $C(\lambda)$ be the set of all corner cells of λ. One can show that

$$f^\lambda = \sum_{x\in C(\lambda)} f^{\lambda - x} \tag{1.2}$$

where $\lambda - x$ is the partition one gets by removing the corner cell x from the Young diagram of λ.

Exercise 1. *Characterize the corner cells in terms of their hook length. (Difficulty rating: 1)*

Exercise 2. *Without using the hook length formula, give a bijective proof of the identity (1.2). (Difficulty rating: 2)*

In light of the identity (1.2), we could try to prove the hook length formula by showing that the RHS of (1.1) follows the same recursive formula. It is clear that both the left- and right-hand side of (1.1) are equal for $\lambda = (1)$ as both sides of (1.1) are equal to 1. So we've reduced our problem to showing that the RHS has the same recursive formula, in particular

$$\frac{|\lambda|!}{\prod_{y\in\lambda} h_\lambda(y)} = \sum_{x\in C(\lambda)} \frac{|\lambda - x|!}{\prod_{y\in\lambda-x} h_{\lambda-x}(y)}. \tag{1.3}$$

Of course, this is a complicated formula involving possible a large sum of large product. So we must be a little clever.

Rearranging (1.3) we find

$$\sum_{x\in C(\lambda)} \frac{1}{|\lambda|} \frac{\prod_{y\in\lambda} h_\lambda(y)}{\prod_{y\in\lambda-x} h_{\lambda-x}(y)} = 1. \tag{1.4}$$

One should notice some simple but important properties of this rearrangement:

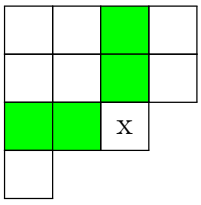
- For each $x \in C(\lambda)$ we assign a positive real number.
- These numbers add up to 1.

Thus we can think of this as giving a probability distribution on the corner cells of λ! The payoff here is that proving that the RHS of (1.1) satisfies the recursive formula (1.3) is equivalent to constructing a probability distribution on the corners of λ in which the probability of choosing a particular corner $x \in C(\lambda)$ is given by

$$\frac{1}{|\lambda|} \frac{\prod_{y\in\lambda} h_\lambda(y)}{\prod_{y\in\lambda-x} h_{\lambda-x}(y)}. \tag{1.5}$$

If we had such a distribution then summing the above over all the corners would be 1, showing that (1.4) is satisfied.

Note that if we fix the corner x, most of the hook length are the same in $\lambda - x$ as they are in λ. The only one which change are those in the *cohook* of x: the cells lying above x in the same column or to the left of x in the same row:



In green, the cohook of the cell x.

If $y \in \text{cohook}(x)$ then $h_{\lambda-x}(y) = h_\lambda(y) - 1$. With this information we can rewrite (1.5) as

$$\begin{aligned} \frac{1}{|\lambda|} \frac{\prod_{y\in\lambda} h_\lambda(y)}{\prod_{y\in\lambda-x} h_{\lambda-x}(y)} &= \frac{1}{|\lambda|} \prod_{y\in\text{cohook}(x)} \frac{h_\lambda(y)}{h_\lambda(y)-1} \\ &= \frac{1}{|\lambda|} \prod_{y\in\text{cohook}(x)} \left(1 + \frac{1}{h_\lambda(y)-1}\right) \end{aligned} \tag{1.6}$$

1.1. **The hook walk algorithm.** Now that we have reformulated the problem as a probability question, let's state an algorithm that does the job.

The hook walk algorithm:

(1) Choose a cell u_0 of λ uniformly at random.

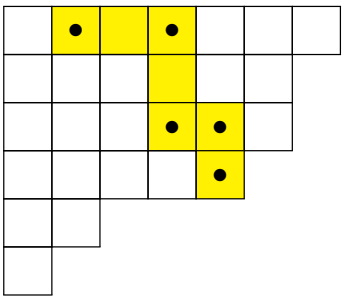
(2) Suppose we are at cell u_i: Pick a new cell u_{i+1} uniformly at random from the cells in the hook of u_i, not including u_i.

(3) Repeat step (2) getting a sequence of cells u_0, u_1, u_2, ..., until the process terminates at a corner cell x.

Proposition 1.2. *Under this algorithm the probability of ending at a given corner cell $x \in C(\lambda)$ is exactly (1.5).*

Let's do an example of this hook walk.

Example 1.3. Let $\lambda = (7, 6, 6, 5, 2, 1)$. We draw a possible hook walk below. The black dots indicate the location of each jump.



In this example we have $u_0 = (1, 2)$, $u_1 = (1, 4)$, $u_2 = (3, 4)$, $u_3 = (3, 4)$, and $x = (4, 5)$.

Suppose we fix a corner x. We have

$$\begin{aligned}\mathbb{P}(\text{hook walk ends at x}) &= \sum_{u_0 \in \lambda} \mathbb{P}(\text{hook walk ends at } x | \text{hook walk starts at } u_0)\mathbb{P}(\text{hook walk starts at } u_0) \\ &= \frac{1}{|\lambda|} \sum_{u_0 \in \lambda} \mathbb{P}(\text{hook walk ends at } x | \text{hook walk starts at } u_0)\end{aligned} \tag{1.7}$$

where we use the fact that we choose our starting cell uniformly at random.

Let consider the sequence of cells $u_0, u_1, \ldots, x$ we reach along our hook walk. Note that if we are at a cell u the probability that we jump to a cell v in its hook is given by

$$\frac{1}{h_\lambda(u) - 1}.$$

Thus the probability of this particular hook walk is

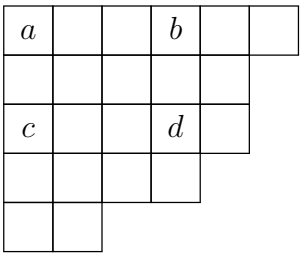
$$\prod_{i \geq 0} \frac{1}{h_\lambda(u_i) - 1}$$

and we have

$$\begin{aligned}\sum_{u_0\in\lambda}\mathbb{P}(\text{h.w. ends at } x|\text{h.w. starts at } u_0) &= \sum_{u_0\in\lambda}\sum_{\substack{(u_0,u_1,\ldots,x)\\ u_0,x \text{ fixed}}}\prod_{i\geq 0}\frac{1}{h_\lambda(u_i)-1}\\ &= \sum_{\substack{(u_0,u_1,\ldots,x)\\ x \text{ fixed}}}\prod_{i\geq 0}\frac{1}{h_\lambda(u_i)-1}\end{aligned} \tag{1.8}$$

We will simplify this using the following lemmas.

Lemma 1.4. *Let a, b, c, and d be four cells of λ that form a rectangle. For example:*



Then we have

$$h_\lambda(a)+h_\lambda(d)=h_\lambda(b)+h_\lambda(c).$$

In particular, if d is a corner cell we have

$$h_\lambda(a)-1=(h_\lambda(b)-1)+(h_\lambda(c)-1).$$

Lemma 1.5. *Fix a corner $x=(r,c)$ of λ, where we indicate the row and column coordinate. For every hook walk we can define the projection sets A and L where A is the set of all columns the hook walk visits and L and the set of all rows. Define $hw(A,L)$ to be the set of all hook walks with the given projection sets A and L.*

$$\sum_{\substack{(u_0,u_1,\ldots,x)\in hw(A,L)\\ x \text{ fixed}}}\prod_{i\geq 0}\frac{1}{h_\lambda(u_i)-1}=\prod_{\substack{a\in A\\ s.t\ a\neq r}}\left(\frac{1}{h_\lambda((a,c))-1}\right)\prod_{\substack{l\in L\\ s.t\ l\neq c}}\left(\frac{1}{h_\lambda((r,l))-1}\right)$$

Proof. This is the main technical lemma we need. We will prove it by induction on the distance from the corner.

Call the probability on the LHS $P(A,L)$. Let's order the elements of A from least to greatest $a_0<a_1<\ldots$ and similarly for the elements of L. Note that the hook walks must start at (a_0,l_0). After one step, the walk could either have jumped right to (a_0,l_1) or down to (a_1,l_0). Both of these jumps happen with equal probability:

$\frac{1}{h_\lambda((a_0,l_0))-1}$. We have

$$\begin{aligned}
P(A,L) &= \frac{1}{h_\lambda((a_0,l_0))-1}\left(P(A-a_0,L)+P(A,L-l_0)\right)\\
&= \frac{1}{h_\lambda((a_0,l_0))-1}\left(\prod_{\substack{a\in A-a_0\\ \text{s.t } a\neq r}}\left(\frac{1}{h_\lambda((a,c))-1}\right)\prod_{\substack{l\in L\\ \text{s.t } l\neq c}}\left(\frac{1}{h_\lambda((r,l))-1}\right)\right.\\
&\qquad\left.+\prod_{\substack{a\in A\\ \text{s.t } a\neq r}}\left(\frac{1}{h_\lambda((a,c))-1}\right)\prod_{\substack{l\in L-l_0\\ \text{s.t } l\neq c}}\left(\frac{1}{h_\lambda((r,l))-1}\right)\right)\\
&= \frac{1}{h_\lambda((a_0,l_0))-1}\left((h_\lambda(a_0,c)-1)\prod_{\substack{a\in A\\ \text{s.t } a\neq r}}\left(\frac{1}{h_\lambda((a,c))-1}\right)\prod_{\substack{l\in L\\ \text{s.t } l\neq c}}\left(\frac{1}{h_\lambda((r,l))-1}\right)\right.\\
&\qquad\left.+(h_\lambda(r,l_0)-1)\prod_{\substack{a\in A\\ \text{s.t } a\neq r}}\left(\frac{1}{h_\lambda((a,c))-1}\right)\prod_{\substack{l\in L\\ \text{s.t } l\neq c}}\left(\frac{1}{h_\lambda((r,l))-1}\right)\right)\\
&= \prod_{\substack{a\in A\\ \text{s.t } a\neq r}}\left(\frac{1}{h_\lambda((a,c))-1}\right)\prod_{\substack{l\in L\\ \text{s.t } l\neq c}}\left(\frac{1}{h_\lambda((r,l))-1}\right)\frac{(h_\lambda(a_0,c)-1)+(h_\lambda(r,l_0)-1)}{h_\lambda((a_0,l_0))-1}\\
&= \prod_{\substack{a\in A\\ \text{s.t } a\neq r}}\left(\frac{1}{h_\lambda((a,c))-1}\right)\prod_{\substack{l\in L\\ \text{s.t } l\neq c}}\left(\frac{1}{h_\lambda((r,l))-1}\right)
\end{aligned}$$

where the first equality follows by induction and the last equality follows by Lemma 1.4. □

Lemma 1.6. *Fix a corner x of λ. Then*

$$\sum_{\substack{(u_0,u_1,\ldots,x)\\ x \text{ fixed}}}\prod_i\frac{1}{h_\lambda(u_i)-1}=\prod_{y\in cohook(x)}\left(1+\frac{1}{h_\lambda(y)-1}\right)$$

where the sum is over all hook walks that end at x.

Proof. First we write the LHS as

$$\sum_{A,L}\sum_{\substack{(u_0,u_1,\ldots,x)\in hw(A,L)\\ x \text{ fixed}}}\prod_i\frac{1}{h_\lambda(u_i)-1}=\sum_{A,L}\prod_{\substack{a\in A\\ \text{s.t } a\neq r}}\left(\frac{1}{h_\lambda((a,c))-1}\right)\prod_{\substack{l\in L\\ \text{s.t } l\neq c}}\left(\frac{1}{h_\lambda((r,l))-1}\right)$$

using Lemma 1.5. Note that each choice of A and L tell us which of the cohooks to pick when we expand out the product on the RHS. □

Now we just need to put everything together. Together with Eqn. (1.8), Lemma 1.6 tells us that

$$\sum_{u_0\in\lambda}\mathbb{P}(\text{h.w. ends at } x|\text{h.w. starts at } u_0)=\prod_{y\in\text{cohook}(x)}\left(1+\frac{1}{h_\lambda(y)-1}\right).$$

Plugging this into Eqn. (1.7), we find that

$$\mathbb{P}(\text{hook walk ends at x}) = \frac{1}{|\lambda|} \prod_{y \in \text{cohook}(x)} \left(1 + \frac{1}{h_\lambda(y) - 1}\right)$$

which we know from Eqns. (1.5) and (1.6) is exactly what we desired. Thus we have proven Proposition 1.2 from which the hook length formula follows.

Exercise 3. *Show that the hook walk algorithm always terminates. (Difficulty rating: 1)*

Exercise 4. *Prove Lemma 1.4. (Difficulty rating: 1)*

Exercise 5. *Suppose λ is a partition of n. Consider the following process for constructing a random standard Young tableaux of shape λ:*

(1) Use the hook walk algorithm to choose a corner cell of λ. Fill that cell with an n and delete it from the Young diagram.

(2) Now use the hook walk algorithm to select another corner cell of the new Young diagram. Fill that cell with an $n-1$ and delete it from the diagram.

(3) Repeat this process until there are no cells left.

Show that this always generates a standard Young tableaux. Show that the tableaux that is generated is chosen uniformly from all possible standard Young tableaux of shape λ. (Difficulty rating: 2)

Exercise 6. *Using Lemma 1.6, show that the probability that a hook walk ends at a corner $x = (r, c)$ given that it starts at $u_0 = (a_0, l_0)$ with $a_0 < r$ and $l_0 < c$ is given by*

$$\left[\frac{1}{h_\lambda((a_0, c)) - 1} \prod_{a_0 < i < r} \left(1 + \frac{1}{h_\lambda((i, c)) - 1}\right)\right] \cdot \left[\frac{1}{h_\lambda((r, l_0)) - 1} \prod_{l_0 < j < c} \left(1 + \frac{1}{h_\lambda((r, j)) - 1}\right)\right]$$

Exercise 7. *Prove that in a rooted tree T (with root at top) with n vertices, the number of ways to label the vertices 1 through n so that every vertex's label is smaller than its parent's label is*

$$\frac{n!}{\prod_{v \in T} h(v)}$$

where $h(v)$ is the number of vertices in the subtree originating from v (including v). (Difficulty rating: 2)

2. Growing Young Diagrams

This method of argument has been applied to many other problems in combinatorics. Here we sketch a proof of the identity

$$\sum_{\lambda \vdash n} (f^\lambda)^2 = n! \tag{2.1}$$

We will follow the ideas of the paper "Another Probabilistic Method in the Theory of Young Tableaux" (1981) by C. Greene, A. Nijenhuis, and H. S. Wilf.

We rewrite (2.1) as

$$\sum_{\lambda \vdash n} \frac{(f^\lambda)^2}{n!} = 1$$

and think of this as giving us a probability measure on the the set of partitions of n such that

$$\mathbb{P}(\lambda) = \frac{(f^\lambda)^2}{n!}.$$

Now recall Eqn. (1.2), which says

$$f^\lambda = \sum_{x \in C(\lambda)} f^{\lambda - x}.$$

Using this we have

$$\begin{aligned} \frac{(f^\lambda)^2}{n!} &= \frac{f^\lambda}{n!} f^\lambda \\ &= \sum_{x \in C(\lambda)} \frac{f^\lambda}{n!} f^{\lambda - x} \\ &= \sum_{x \in C(\lambda)} \frac{f^\lambda}{n f^{\lambda - x}} \frac{(f^{\lambda - x})^2}{(n-1)!} \end{aligned} \tag{2.2}$$

We interpret this as follows: Suppose we have a probability distribution on partitions λ' of $n-1$. Say we have a Markov chain that tells us how to our partitions λ' of $n-1$ into partitions λ of n with transition probabilities $\mathbb{P}(\lambda' \to \lambda)$. Then this gives us a probability distribution of the partitions of n through

$$\mathbb{P}(\lambda) = \sum_{\lambda' \vdash n-1} \mathbb{P}(\lambda' \to \lambda) \mathbb{P}(\lambda').$$

Suppose we know that (2.1) holds for partitions λ' of $n-1$, and so can choose $\mathbb{P}(\lambda') = \frac{(f^{\lambda'})^2}{(n-1)!}$. Next let's assume that we can choose the transition probabilities to be

$$\mathbb{P}(\lambda' \to \lambda) = \begin{cases} \frac{f^\lambda}{n f^{\lambda'}} & \text{if you can delete a single corner of } \lambda \text{ to get } \lambda' \\ 0 & \text{o.w.} \end{cases}$$

Then the calculation in (2.2) tells us that the probability measure we get on partitions of n is exactly

$$\mathbb{P}(\lambda) = \frac{(f^\lambda)^2}{n!}.$$

Since this is a probability distribution, summing over all λ must give us 1, and this implies Eqn. (2.1).

All of this works as long as our transition probabilities are well-defined. They are clearly positive real numbers, so it is enough to check

$$\sum_{\lambda \,:\, \lambda' \to \lambda} \frac{f^\lambda}{n f^{\lambda'}} = 1 \tag{2.3}$$

where $\lambda' \to \lambda$ means that we can add a single corner cell to λ' to get λ. Fixing λ', we would like to construct a variant of our hook walk that chooses to add the cell x to λ' with probability

$$\frac{f^{\lambda'+x}}{n f^{\lambda'}} = \frac{\prod_{y \in \lambda'} h_{\lambda'}(y)}{\prod_{y \in \lambda'+x} h_{\lambda'+x}(y)} \tag{2.4}$$

where we used the hook length formula. Note that we want to add a cell to λ', that is, we are growing our partition.

2.1. **Complementary Hook Walk.** Here we will describe the hook walk that does the job and leave the details of the proof to the reader.

The complementary hook walk algorithm

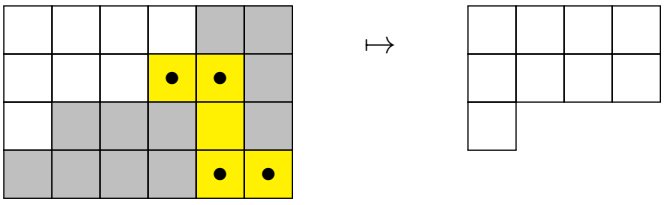
(1) Embed λ' into a rectangle R of size $p \times q$ with p larger than the first column of λ' and q larger than the first row. Note that $R - \lambda'$ is itself the Young diagram of some partition after rotating 180 degrees.

(2) Do a hook walk on $R - \lambda'$ starting from the cell $(1, 1)$ (equivalently (p, q) in the non-rotated picture).

(3) The hook walk will terminate at a corner of $R - \lambda'$. Add this cell to λ' to get λ.

Let's do an example to make this clear.

Example 2.1. Let $\lambda' = (4, 3, 1)$. Embed this partition in the rectangle R of length 4×6. Note $R - \lambda'$, after rotating, is the partition $(6, 5, 3, 2)$. We show a possible complementary hook walk below. The black dots indicate the location of the jumps.



We add the terminal cell of the hook walk to our partition and get $\lambda = (4, 4, 1)$.

Exercise 8. *Prove that the complementary hook walk adds cells to λ' with the desired probability* (2.4)*. (Difficulty rating: 3).*

The probability distribution $\mathbb{P}(\lambda) = \frac{(f^\lambda)^2}{n!}$ is known as the *Plancherel measure*. It is interesting in its own right and has relations to many branches of mathematics. Note that by starting with the empty partition and repeatedly using the complementary hook walk, we can grow a large Plancherel-distributed random partition. Such partitions exhibit an interesting limit shape phenomenon. See the book "The Surprising Mathematics of Longest Increasing Subsequences" by Dan Romik for an a nice exposition of these topics.

3. Oscillating tableaux

In this section we combine the two types of hook walks we discussed previously. We follow "Area statistics for large oscillating tableaux" (2020) D. Keating.

An oscillating tableaux of length N and shape λ is a sequence of partitions

$$\{\lambda^{(0)} = \emptyset, \lambda^{(1)}, \ldots, \lambda^{(N-1)}, \lambda^{(N)} = \lambda\}$$

such that either $\lambda^{(i)} \to \lambda^{(i+1)}$ or $\lambda^{(i+1)} \to \lambda^{(i)}$. That is, to go from the i-th to the $(i+1)$-th partition you add or delete a corner. We denote the set of all oscillating tableaux of length N and shape λ by $\mathcal{OT}(\lambda, N)$. It is a well-known fact that the number of such tableaux for fixed shape and length has a very simple formula. Let λ be a partition with $|\lambda| = k$. Then for all $n \in \mathbb{N}$ let $N = k + 2n$, we have

$$f_N^\lambda = \#\mathcal{OT}(\lambda, N) = \binom{N}{k}(N-k-1)!!f^\lambda \tag{3.1}$$

where f^λ is the number of standard Young tableaux of shape λ. Further, $\#\mathcal{OT}(\lambda, l) = 0$ if $l \neq k + 2n$ for some $n \in \mathbb{N}$.

Example 3.1. As an example, here are all oscillating tableaux of shape $\lambda = \emptyset$ and length 4.

$$\emptyset \to (1) \leftarrow \emptyset \to (1) \leftarrow \emptyset$$
$$\emptyset \to (1) \to (1,1) \leftarrow (1) \leftarrow \emptyset$$
$$\emptyset \to (1) \to (2) \leftarrow (1) \leftarrow \emptyset$$

We would like to give a proof of Eqn. (3.1). First, we will a recursion relation for the oscillating tableaux.

Lemma 3.2.

$$f_N^\lambda = \sum_{\mu \to \lambda} f_{N-1}^\mu + \sum_{\lambda \to \mu} f_{N-1}^\mu$$

Exercise 9. *Prove Lemma 3.2 without using the formula in (3.1). (Difficulty rating: 1)*

If we can show that the formula on the RHS of (3.1) satisfies the same recursion relation then we are done (modulo the base case). Showing this it is equivalent to showing

$$\frac{k}{N}\sum_{\mu\,:\,\mu\to\lambda}\frac{f^\mu}{f^\lambda} + \frac{N-k}{N}\sum_{\mu\,:\,\lambda\to\mu}\frac{f^\mu}{(k+1)f^\lambda} = 1$$

We can think of this as a probability distribution on the corners that can be added or removed for λ. In particular, the probability of adding a corner x is

$$\frac{N-k}{N}\frac{f^{\lambda+x}}{(k+1)f^\lambda}$$

while the probability of removing a corner x is

$$\frac{k}{N}\frac{f^\mu}{f^\lambda}.$$

If we can construct an algorithm that selects corners with these probabilities, then we would know that they sum to 1, and the recursion would be satisfied.

3.1. **The oscillating hook walk algorithm.** Comparing the probabilities above to what we have covered previously suggests the following algorithm.

The oscillating hook walk algorithm:

(1) Choose to add a corner with probability $\frac{N-k}{N}$ or remove a corner with probability $\frac{k}{N}$.

(2) (a) If you chose to add a corner, pick which corner to add by doing a complementary hook walk.

(b) If you chose to remove a corner, pick which corner to remove by doing a hook walk.

In fact, the proof that this has gives the desired distribution on the possible corners to add or remove is almost immediate from what we know know about the hook walk and the complementary hook walk.

Exercise 10. *Prove the oscillating hook walk algorithm gives the desired probabilities. (Difficulty rating: 1)*

Iterating this, we get a nice algorithm for generating uniformly random oscillating tableaux.

(1) Fix N and λ such that $|\lambda| = k$ and $N = k + 2n$ for some $n \in \mathbb{N}$.

(2) Set $\lambda^{(N)} = \lambda$. Set $X = 0$. Set $Y = k$.

(3) While $X < N$:

(a) With probability $\frac{Y}{N-X}$, remove a corner from $\lambda^{(N-X)}$ using the hook-walk algorithm. Set the new partition to $\lambda^{(N-X-1)}$. Take $X \to X + 1$, $Y \to Y - 1$.

(b) Otherwise, add a corner to $\lambda^{(N-X)}$ using the complementary hook-walk algorithm. Set the new partition to $\lambda^{(N-X-1)}$. Take $X \to X + 1$, $Y \to Y + 1$.

(4) Return the oscillating tableaux $\{\lambda^{(0)} = \emptyset, \lambda^{(1)}, \ldots, \lambda^{(N-1)}, \lambda^{(N)} = \lambda\}$.

Exercise 11. *(1) Write a program the generates uniformly random standard Young tableaux of shape λ using the hook walk algorithm.*

(2) Write a program the generates Plancherel-distributed partitions of n using the complementary hook walk algorithm.

(3) Write a program that generates uniformly random oscillating tableaux of shape λ and length N using the oscillating hook walk algorithm.

(Difficulty rating: 2)