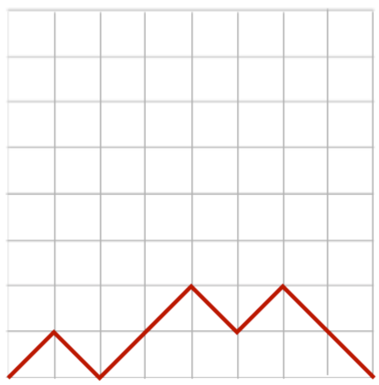
Ranking and unranking Dyck paths

Contents

1 Ranking and unranking Dyck paths

1.1 Dyck paths to words

Recall Dyck paths from lecture 5



We can represent a Dyck path as a word using the letters $\{\nearrow, \searrow\}$. If we write $\nearrow$ as 0 and $\searrow$ as 1 we can represent a Dyck path as a binary string.

Which binary strings do we get?

Definition. Say a binary string of length $2n$ is totally balanced if

- The string contains n zeros and n ones.
- For any $1 \leq i \leq 2n$, the first i elements of the string include at least as many zeros and ones.

This is exactly the same as the condition defining Dyck paths: the second point is that the path never goes strictly below the x-axis, while the first point is that the path returns to the x-axis at the end.

Viewing Dyck paths as binary strings gives them a natural lexicographic order. This is the order we will rank and unrank with respect to.

1.2 Counting suffixes of Dyck paths

Definition. Let $\mathcal{D}_{2n}(x, y)$ be the set of paths from (x, y) to $(2n, 0)$ using the steps $(1, 1)$ and $(1, -1)$ and which never go strictly below the x-axis.

Let $d_{2n}(x, y) = |\mathcal{D}_{2n}(x, y)|$.

Such paths are ends (suffixes if you think of them as words) of Dyck paths.

Proposition. *Let x,y, and n be positive integers with $x + y$ even and $x + y \leq 2n$. Then*

$$d_{2n} = \binom{2n-x}{n - \frac{x+y}{2}} - \binom{2n-x}{n-1-\frac{x+y}{2}}$$

Proof. Lets first count all paths using the steps $(1,1)$ and $(1,-1)$ which go from (x,x) to $(2n,0)$ without regards to whether or not they cross the x-axis.

Such paths must contain $2n-x$ steps, since they need to move $2n-x$ units in the x direction. Furthermore such paths must move a net of y steps in the y direction. So there are y *extra* $(1,-1)$ steps, and the remaining steps are evenly divided between $(1,1)$ and $(1,-1)$. So there are $2n-x-y$ such remaining steps, half of each type. Thus there are

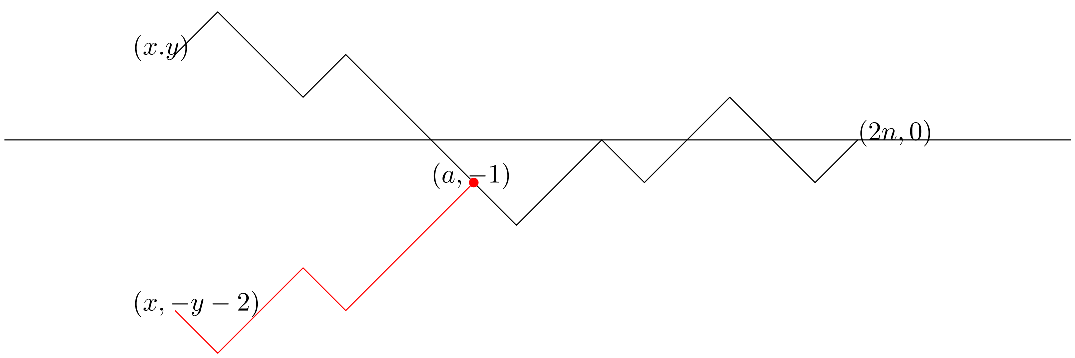
$$n-\frac{x+y}{2}\ (1,1)\text{ steps}$$
$$n-\frac{x+y}{2}+y\ (1,-1)\text{ steps}$$

and no restriction on which order the steps come in. So there are

$$\binom{2n-x}{n-\frac{x+y}{2}}$$

such paths.

Now we just need to subtract off those paths which do go strictly below the x axis. Let w be such a path, and let $(a,-1)$ be the integer point where it is first strictly below the x-axis. Flip the portion of the path before $(a,-1)$ across the line $y=-1$, as illustrated



Note that the modified path goes from $(x,-y-2)$ to $(2n,0)$. Note also that given a path from $(x,-y-2)$ to $(2n,0)$ there is a unique point where it may have been flipped according to this rule: namely the first place where the path reaches $(a,-1)$ for some a.

Thus the number of paths from (x,y) to $(2n,0)$ which do go strictly below the x-axis is the same as the number of all paths from $(x,-y-2)$ to $(2n,0)$. Counting as before this is

$$\binom{2n-x}{n-1-\frac{x+y}{2}}$$

Therefore the number of paths from (x,y) to $(2n,0)$ which never go strictly below the x-axis is

$$\binom{2n-x}{n-\frac{x+y}{2}}-\binom{2n-x}{n-1-\frac{x+y}{2}}$$

□

1.3 Ranking and unranking

The formula for counting suffixes tells us about the rank. Suppose $w=w_1w_2\cdots w_{2n}$ is a Dyck path represented as a binary string. Suppose that after the first k steps the path is at the point (x,y). If $w_{k+1}=1$ so the next step is down, then all the words which begin $w_1w_2\cdots w_k0$ come before w in lexicographic order. There are $d_{2n}(x+1,y+1)$ such paths (the $+1$s come from the last up step). Thus we have

```
Algorithm: RankDyck
input: n, w.   w is a totally balanced word of length 2n
y=0
r=0
for x from 1 to 2n
  if w(x) = 0
     y=y+1
  else
     r=r+binom(2n-x,n-(x+y)/2)-binom(2n-x,n-1-(x+y)/2)
     y=y-1
output: r
```

```
Algorithm: UnrankDyck
input: n,r.
y=0
rlow=0
for x from 1 to 2n
  m = binom(2n-x,n-(x+y)/2)-binom(2n-x,n-1-(x+y)/2)
  if r <= rlow + m - 1
     y=y+1
     w(x)=0
  else
     rlow = rlow + m
     y=y-1
     w(x)=1
output: w
```

Here's an example of each algorithm which comes from Kreher and Stinson, *Combinatorial Algorithms*, section 3.4.

Suppose we want to compute $rank(w = 0010110101)$. Then:

x	$w(x)$	y	r
1	0	1	0
2	0	2	0
3	1	1	14
4	0	2	14
5	1	1	18
6	1	0	21
7	0	1	21
8	1	0	22
9	0	1	22
10	1	0	

so the rank is 22.

Now to calculate $unrank(22)$

m	x	y	$rlow$	$w(x)$
$d_{10}(1,1) = 42$	1	1	0	0
$d_{10}(2,2) = 28$	2	2	0	0
$d_{10}(3,3) = 14$	3	1	14	1
$d_{10}(4,2) = 9$	4	2	14	0
$d_{10}(5,3) = 4$	5	1	18	1
$d_{10}(6,2) = 3$	6	0	21	1
$d_{10}(7,1) = 2$	7	1	21	0
$d_{10}(8,2) = 1$	8	0	22	1
$d_{10}(9,1) = 1$	9	1	22	0
$d_{10}(10,2) = 0$	10	0	22	1

so we unrank to 0010110101 as expected.