Summary on Lecture 15, February 17, 2016

More on Rooted Trees

Let $m \geq 1$. Recall that a rooted tree (T, r) is a *complete m-ary tree* if every vertex of T has either m children or no children. Mostly we are interested in the case $m = 2$.

Lemma 1. *Let (T, r) be a complete binary tree. Then $|V(T)|$ is odd.*

Exercise. Prove Lemma 1 by induction.

We would like to count how many complete binary trees are there with $2n + 1$ vertices.

Let (T, r) be a complete binary tree with $2n + 1$ vertices. We use preorder listing to give a a list of all vertices (starting with the root): $rv_1v_2 \dots v_{2n}$. We notice that every move from v_i to v_{i+1} has a direction: its either left (L) or right (R). Hence the list $rv_1v_2 \dots v_{2n}$ gives a sequence of $2n$ L's and R's. Then we notice:

- We visit first the "left" child, then the "right" one. Thus if we count how many L's and R's from the beginning to a given spot, we'll get that the number of L's is greater or equal to the number of R's.
- There are n L's and n R's.

We have seen this problem before, and conclude that the number of such listings (and, consequently, the number of complete binary graphs with $2n + 1$ vertices) is nothing but the *Catalan number*, namely, $\frac{1}{n+1}\binom{2n}{n}$.

Recall definition of the Catalan numbers. Let us consider the xy-plane, and two types of moves:

$$\mathsf{R} : (x, y) \mapsto (x + 1, y), \quad \mathsf{U} : (x, y) \mapsto (x, y + 1).$$

We are allowed to make the moves R and U to get from the point $(0, 0)$ to the point (n, n). A path consisting of only the moves R and U is called **monotonic**.

Warm-up question: How many monotonic paths are there from $(0, 0)$ to (n, n)?

This is easy. Indeed, any monotonic path can be recorded as a sequence of n R's and n U's. A total number of moves is $2n$; thus it is enough to choose n slots for R's (or n U's). We obtain $\binom{2n}{n}$ paths.

A monotonic path from $(0, 0)$ to (n, n) is **dangerous** if it crosses the diagonal.

Actual question: How many non-dangerous monotonic paths are there from $(0, 0)$ to (n, n)?

Let $n = 6$. Then the paths

R R U R U U R U R U R U is non-dangerous,

R R U R U U R U U U R R is dangerous.

To distinguish dangerous and non-dangerous paths, we count how many R and U moves did we make at every step:

10 20 21 31 32 33 43 44 54 55 65 66
R R U R U U R U R U R U is non-dangerous,

⇓
10 20 21 31 32 33 43 44 45 46 56 56
R R U R U U R U U U R R is dangerous.

Moreover, once the number of U-moves gets greater than the number of R-moves, we use the red color. Then, once the first red indicator appears, we write new path, where we change the path after the dangerous U-move:

all R-moves we turn to U-moves, and all U-moves we turn to R-moves:

								⇓				
10	20	21	31	32	33	43	44	45	46	56	56	
R	R	U	R	U	U	R	U	U	U	R	R	a dangerous path.

								⇓		
10	20	21	31	32	33	43	44	45		
R	R	U	R	U	U	R	U	U	R U U	new path.

In the black portion of the new path, we have 4 R-moves and 5 U-moves; in the red portion, we have 1 R-move and 2 U-moves. Totally, new path has 5 R-moves and 7 U-moves. Thus it is a path from $(0,0)$ to $(5,7)$. We claim that in this way every dangerous path turns to a path from $(0,0)$ to $(5,7)$. Thus we have the answer:

$$\{\# \text{ of all paths}\} - \{\# \text{ of dangerous paths}\} = \binom{12}{6} - \binom{12}{5}.$$

For general n, we do the same. Namely, we consider a dangerous path (first line) and we produce new path below:

	⇓	
$(k-1)$ U's, $(k-1)$ R's	U	$(n-k)$ U's, $(n-k+1)$ R's
$(k-1)$ U's, $(k-1)$ R's	U	$(n-k)$ R's, $(n-k+1)$ U's

The first path is dangerous since the red marker ⇓ shows that there are k U's and $(k-1)$ R's, so the path crossed the diagonal. For the new path we changed all U's by R's and all R's by U's **after** the red marker ⇓. Totally, for the new path, we have

$$\begin{array}{ll} k+n-k+1 = n+1 & \mathsf{U}'s \\ k-1+n-k = n-1 & \mathsf{R}'s \end{array}$$

Thus we have the answer:

$$b_n := \binom{2n}{n} - \binom{2n}{n-1} = \frac{1}{n+1}\binom{2n}{n}.$$

Now we return to trees. Let $G = (V, E)$ be a connected graph without loops and multiple edges. We assume that the vertices of G are ordered, i.e., $V = \{v_1, \ldots, v_n\}$. We would like to find a spanning tree (T, r) (which is *depth-first ordered rooted tree*).

Here is a pseudocode for a recursive version of the Depth-First-Search algorithm:

Depth-First-Search (G, v)

```
    Let v := v1 .  Put v to the list T
    For all edges from v to w in E(G) do
        if w is not in T then call T(G,w) := Depth-First-Search(G,w) ,
        T := T ∪ T(G,w)
    Return T
```

Exercise. Use **Depth-First-Search** (G, v) algorithm for several large graphs. Find non-trivial examples.

Exercise. Study the **Breadth-First-Search** (G, v) algorithm from the textbook and write a pseudocode for its recursive version.