Math 180 Homework 5

Jiaping Zeng

11/20/2020

8.3.1 **Answer**: Each component of such a graph is a directed cycle since by construction, each vertex has exactly one edge going in and one edge going out, therefore forming a directed cycle by definition. Since our original graph is a tree, it does not contain any cycle, therefore each component will have only the constructed directed cycle and no other cycles. Then, the branches attached to the chord in the original graph are simply attached onto the cycles, with the edges directed towards the cycle by construction.

8.3.2 **Answer**: We can reconstruct the chord first by looking at the vertices in the cycles; we can first order the vertices by their labels, i.e. $v_i, v_{i+1}, v_{i+2}, \ldots$, then construct a new list by finding the vertex each vertex in our ordered list is connected to. This gives us the chord. Then for the branches, we can simply take them from the cycle and reattach to the chord, and make them undirected.

8.4.1 **Answer**: Since T is a tree, we have $\sum_i m_i = n-2$. In addition, we also have $|E| = n-1$, so $\sum_i(\deg_T(i)-1) = 2|E| - n = 2(n-1) - n = n-2$. Then $\sum_i m_i = \sum_i(\deg_T(i)-1)$, so we must have $m_i = \deg_T(i) - 1 \implies \deg_T(i) = m_i + 1$ for all i.

P10 Use the Prüfer code to count the number of spanning trees of K_n with $\deg(v_1) = 2$.

Answer: Since $\deg(v_1) = 2$, we have $m_1 = 1$ by problem 8.4.1, i.e. v_1 shows up only once in the Prüfer code. Then since K_{n-1} has $(n-1)^{n-3}$ spanning trees by Cayley's formula, it also has $(n-1)^{n-3}$ unique Prüfer codes. We can place v_1 anywhere in the Prüfer codes (each has length $n-2$), so there are $(n-2)(n-1)^{n-3}$ such spanning trees.

P11 Use the Prüfer code to count the number of spanning trees of K_n with each vertex degree 1 or 3.

Answer: By problem 8.4.1, spanning trees have $\deg_T(i) = m_i + 1$ for all i, so our spanning trees must have $m_i = 0$ or $m_i = 2$. In other words, each vertex in the code shows up exactly twice. Then, since $\sum_i m_i = n-2$, exactly $\frac{n-2}{2}$ vertices are in the code. Then we have $(n-2)!$ ways to arrange the vertices, with $2^{\frac{n-2}{2}}$ duplicates for each code. Therefore we have $\binom{n}{\frac{n-2}{2}} \cdot \dfrac{(n-2)!}{2^{\frac{n-2}{2}}}$ spanning trees in K_n.