Practice Session Problems
CS 273 Introduction to Theoretical Computer Science
Spring Semester, 2004

held Tuesday, April 13

1. Find a tight asymptotic bound for the following recurrence:
   \[ T(n) = 5T\left(n^{1/5}\right) + \lg n \]
   Solution:
The \( n^{1/5} \) term makes things nasty. We are more used to stuff when we divide \( n \) by some constant. To make it that form, we could substitute \( n \) with something that is the power of a constant, say \( n = c^k \), and we would get \( n^{1/5} = c^{k/5} \). So, if we defined another function \( S(k) \) that satisfied \( S(k) = T(c^k) \), then the recurrence would look like \( S(k) = 5S(k/5) + \lg c^k \). To make \( \lg c^k \) pretty, we should choose the constant \( c \) to be 2, to get \( \lg c^k = k \). So our recurrence is now \( S(k) = 5S(k/5) + k \).

Using the Master's theorem would be nice, but you can never remember it when you need to (like in an exam). You recall the general form of a divide-and-conquer recurrence was \( R(k) = aR(k/b) + f(k) \). If you keep recursively using this formula, you get \( R(k) = a(aR(k/b^2) + f(k/b)) + f(k) = a^2(aR(k/b^3) + f(k/b^2)) + f(k/b) + f(k) = \ldots \). So eventually you get something like \( R(k) = a^k f(1) + \cdots + a^2 f(k/b^2) + af(k/b) + f(k) \). \( L \) should be something that makes \( k/b^L = 1 \), which is \( \log_b k \).

You can plug in the \( S(k) \) we got above for \( R(k) \) and work the summation, but it does take some time. We recall from the proof of Master's theorem that we compared the ratio between the terms in the sum to simplify the summation. So we try comparing \( af(k/b) \) with \( f(k) \). For \( S(k) \), \( af(k/b) = k \) and \( f(k) = k \). So, we can see that all the terms in the sum are equal. So the sum is \( L \times k = \log_b k \times k = \Theta(k \lg k) \). Since \( n = 2^k \), \( T(n) = \lg n \log \lg n \).

2. A tournament is a directed graph \( G = (V,E) \) such that for all \( u,v \in V \), \( u \neq v \) either \( (u,v) \in E \) or \( (v,u) \in E \). Show that there is a hamiltonian path in \( G \). In other words, show that there exists a permutation \( v_1,v_2,\ldots,v_n \) of vertices in \( V \) such that \( (v_i,v_{i+1}) \in E \) for \( i = 1,2,\ldots,n-1 \).
   Solution:
   We can try to use proof by contradiction, but it doesn’t seems that feasible, since the assumption we would make is “there is no hamiltonian path in \( G \)”. It is hard to start some string of assertions from the fact that something does not exist.

   The only other tool we know is induction. We can induct over a variety of quantities (number of vertices, edges, cycles, etc.), but vertices seem to make more sense here, especially since the number of edges is determined by the number of vertices (a tournament without the directions in the edges would be a complete graph).

   Let’s assume that the proposition holds with graphs that have \( n \) vertices or less. We need to show that it also holds for any graph \( G \) with \( n+1 \) vertices. In this graph, pick some vertex and call it \( u \). The subgraph of \( G \) with just \( u \) omitted is an \( n \)-vertex tournament itself. Hence it has some Hamiltonian path: let’s label the vertices in this path \( v_1,v_2,\ldots,v_n \), where there is a directed edge from \( v_i \) to \( v_{i+1} \) for all \( i = 1,\ldots,n-1 \). If \( (u,v_1) \in E \) or \( (v_n,u) \in E \), we are done, since we can either append or prepend \( u \) to this path.
If not, \((v_1, u) \in E\) and \((u, v_n) \in E\). Then, if \((u, v_2) \in E\), there is a path passing vertices in the order \(v_1, u, v_2, v_3, \ldots, v_n\). If not, then \((v_2, u) \in E\), and hence if \((u, v_3) \in E\), then there is a path that reads \(v_1, v_2, u, v_3, \ldots, v_n\). In this fashion, if \((v_i, u) \in E\) for \(i = 1 \ldots k\) and \((u, k + 1) \in E\), then there is a path \(v_1, v_2, \ldots, v_k, u, v_{k+1}, \ldots, v_n\). Such a \(k\) must exist since \((u, v_n) \in E\).

3. Let \(G\) be a graph with \(n\) vertices and \(e\) edges. Let \(\delta\) be the minimum degree of a vertex in \(G\); let \(\Delta\) be the maximum degree of a vertex in \(G\). Prove that \(\delta \leq 2e/n \leq \Delta\).

**Solution:**

A typical thing that sits between the minimum and maximum is the average. The average degree of a graph would be the sum of all degrees divided by \(n\). The sum of all degrees is twice the number of edges, since each edge meets with two vertices. Hence, the average degree would be \(2e/n\), which is exactly what we want.

4. Let \(a\) be a symbol, \(L\) be regular. Prove that the set of strings of \(L\) that start with \(a\) with that \(a\) removes is also regular.

**Solution:**

Since \(L\) is regular, it has a DFA, \(D\) that recognizes it. \(D\) will have a single state \(q\) that it goes to after reading \(a\) as its first character. Let \(D'\) be a DFA that is identical to \(D\), except that its start state is \(q\). We want to say that \(D'\) recognizes the target language. In other words, we claim that \(aw\) is accepted by \(D\) if and only if \(w\) is accepted by \(D'\). Observe in \(D\) that after one transition in reading \(aw\), \(D\) will be in the same state of \(D'\) before \(D'\) starts reading \(w\). The two machines transition tables are identical, and the input left to read is also identical, so they will end up in the same state after reading all the input. Therefore, \(D\) accepts if and only if \(D'\) accepts.