Assignment 2

due Friday, January 31, 2020

1. exercise 12, p 112
2. exercise 7, p 191-192
3. exercise 15, p 196

4. In class we spoke about the uses of a maximum matching to some other settings. Here is one: suppose we have a collection of \( n \) bricks \( B_1, B_2, \ldots, B_n \). We know that we can stably set one brick \( B_i \) on top of another brick \( B_j \) based on their dimensions. \( B_i \) would have to be sufficiently “smaller” then \( B_j \) for it to be stable - this will be built into a function \( \text{STACKONTOP}(i, j) \), which returns true iff \( B_i \) can be safely (=stably) put on top of \( B_j \). Note that

- \( \text{STACKONTOP}(i, i) \) is false
- if \( \text{STACKONTOP}(i, j) \) is true, then \( \text{STACKONTOP}(j, i) \) is false (on the other hand, they could both be false)
- if \( \text{STACKONTOP}(i, j) \) and \( \text{STACKONTOP}(j, k) \) are true, then \( \text{STACKONTOP}(i, k) \) is true

Our goal is to determine the smallest number \( 1 \leq k \leq n \) of piles such that we can stack all \( n \) bricks into \( k \) piles. A pile of bricks can be as high as \( n \), and each brick must be stable on top of the one below it (there is no restriction for the one on the bottom).

To this end, we construct a bipartite graph \( G \) consisting of vertices \( V = L \cup R \) where \( L = \{b_1, b_2, \ldots, b_n\} \) and \( R = \{b'_1, b'_2, \ldots, b'_n\} \). The edge set consists of all \( (b_i, b'_j) \) where \( \text{STACKONTOP}(i, j) = \text{true} \). Suppose that \( m \) is the size of a maximum matching in \( G \) (you do not need to give an algorithm that determines \( m \)). Show how to determine \( k \) from \( m \), and justify this choice.