CS 473 ✧ Spring 2020

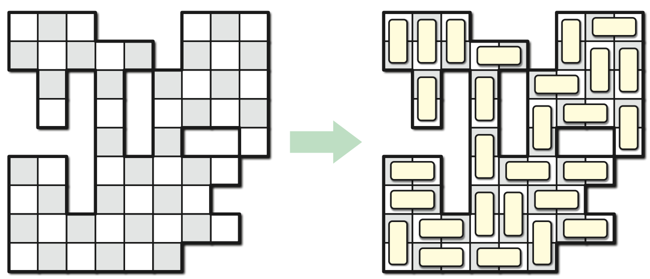
Homework 8

Due Wednesday, April 15, 2020 at 9pm

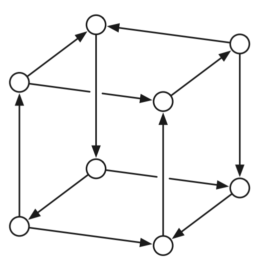
---

1. Suppose you are given an $n \times n$ checkerboard with some of the squares deleted. You have a large set of dominos, just the right size to cover two squares of the checkerboard. Describe and analyze an algorithm to determine whether one tile the board with dominos—each domino must cover exactly two undeleted squares, and each undeleted square must be covered by exactly one domino.

   Your input is a boolean array $\mathit{Deleted}[1..n, 1..n]$, where $\mathit{Deleted}[i, j] = \textsc{True}$ if and only if the square in row i and column j has been deleted. Your output is a single boolean; you do ***not*** have to compute the actual placement of dominos. For example, for the board shown below, your algorithm should return $\textsc{True}$.



2. A ***k-orientation*** of an undirected graph G is an assignment of directions to the edges of G so that every vertex of G has at most k incoming edges. For example, the figure below shows a 2-orientation of the graph of the cube.



   Describe and analyze an algorithm that determines the smallest value of k such that G has a k-orientation, given the undirected graph G input. Equivalently, your algorithm should find an orientation of the edges of G such that the maximum in-degree is as small as possible. For example, given the cube graph as input, your algorithm should return 2.

3. Suppose you have a sequence of jobs, indexed from 1 to i, that you want to run on two processors. For each index i, running job i on processor 1 requires $A[i]$ time, and running job i on processor 2 takes $B[i]$ time. If two jobs i and j are assigned to different processors, there is an additional communication overhead of $C[i,j] = C[j,i]$. Thus, if we assign the jobs in some subset $S \subseteq \{1,2,\dots,n\}$ to processor 1, and we assign the remaining $n-|S|$ jobs to processor 2, then the total execution time is

$$\sum_{i \in S} A[i] \;+\; \sum_{i \notin S} B[i] \;+\; \sum_{i \in S} \sum_{j \notin S} C[i,j].$$

Describe an algorithm to assign jobs to processors so that this total execution time is as small as possible. The input to your algorithm consists of the arrays $A[1..n]$, $B[1..n]$, and $C[1..n, 1..n]$.