CSE 565 **Martin Fürer** **Algorithm Design and Analysis**

Assignment 1 Given: 8/29/18 Due: Wednesday 9/5/18

Exercises

Exercises are for your own practice. Don't hand them in.

1. Solve Exercise 7, Chapter 2 on page 69 of the Textbook.
(Complexity of songs)

2. Solve Exercise 1, Chapter 3 on page 107 of the Textbook.
(Number of topological orders)

3. Solve Exercise 4, Chapter 3 on page 107 of the Textbook.
(Bipartite with same or different)

Problems

Problem solutions have to be handed in. A subset of them will be graded.

Remember: Even though collaboration is permitted, you have to write the solutions by yourself without assistance. Be ready to explain them orally if asked. Write your collaborators on your first sheet. You are not allowed to get solutions from outside sources such as the Web or students not enrolled in this class.
For additional information see the Course Announcement on Canvas.

1. [15 points] Solve Problem 6, Chapter 2 on page 68 of the Textbook.
(Efficient computations of sums)

2. [10+10=20 points] Solve Problem 8, Chapter 2 on page 69 of the Textbook.
(Combining two strategies)

3. [10 points] Solve Problem 3, Chapter 3 on page 107 of the Textbook.
(DAG or cycle)

4. [10+5=15 points] (Types of edges in DFS)
When DFS (depth-first search) traverses a directed graph G, there are three types of visits of every node v of G. The first visit of v is the pre-visit. Then DFS visits all subtrees rooted at the children of v. Inbetween, it in-visits v. Finally, coming back from the last subtree, DFS does the post-visit of v. If v is a leave, we consider the post-visit to happen immediately after the pre-visit. DFS can easily

be augmented with a counter that starts at 0 and is increased during each pre-visit and post-visit. When v is pre-visited, then the value of the counter is copied to $A[v]$. When v is post-visited, then the value of the counter is copied to $B[v]$. Here, A and B are arrays indexed by the vertices. The values are timestamps of the visits. Initially, A and B have the value ∞.

(a) Assume DFS is at a vertex u and considers an adjacent vertex v. How is the edge $\{u, v\}$ classified as a Tree edge, Forward edge, Back edge, or Cross edge depending on the current entries of $A[u], B[u], A[v], B[v]$?

(b) How can DFS decide wether G has a directed cycle?