Instructions

Please read these instructions and all questions C A R E F U L L Y before proceeding.

You will have 2\frac{1}{2} hours. The exam is closed-book. All questions are of equal weight. You must attempt to answer all of them to earn an unconditional pass. All algorithms should be accompanied by a proof of correctness and complexity analysis unless otherwise directed.

The questions will be graded according to the following criteria in order of importance:

1. correctness and completeness
2. clarity, precision, and conciseness
3. optimality of result

Partial credit will be awarded where appropriate, so be sure to show all work. Unclear or irrelevant arguments will be penalized.

Good luck!
1. A numeric function $f : \mathbb{N} \to \mathbb{N}$ with finite range is called binary automatic if there exists a finite set $C$, a function $F : C \to \mathbb{N}$, and a finite automaton on states $C$ with start state $s$ and deterministic transition function $\delta : C \times \{0, 1\} \to C$ such that for all $x \in \{0, 1\}^*$,

$$f(\#x) = F(\delta(s, x)),$$

where $\#x$ denotes the number represented by $x$ in binary (leading zeros permitted). Here $\hat{\delta} : C \times \{0, 1\}^* \to C$ is the multistep extension of $\delta$ defined by $\hat{\delta}(q, \varepsilon) = q$ and $\hat{\delta}(q, xa) = \delta(\delta(q, x), a)$ for $x \in \{0, 1\}^*$, $a \in \{0, 1\}$, $q \in C$, and $\varepsilon$ is the null string.

For example, the function $n \mapsto n \mod 3$ is binary automatic using $C = \{0, 1, 2\}$, $F(q) = q$, $s = 0$, and

$$\delta(q, a) = 2q + a \mod 3$$

for $q \in \{0, 1, 2\}$ and $a \in \{0, 1\}$.

Let $m$ be an arbitrary but fixed positive integer.

(a) Show that any function of the form $n \mapsto f(n) \mod m$, where $f$ is a polynomial with integer coefficients, is binary automatic.

(b) Show that $n \mapsto 2^n \mod m$ is binary automatic.

(c) Give a specific example of a function $\mathbb{N} \to \mathbb{N}$ with finite range that is not binary automatic, and prove that it is not.

2. Suppose we have $n$ jobs to be scheduled on $m$ machines. Each job consists of a set of subtasks, where each subtask takes one unit of time and must be scheduled on a specified machine. Each machine can process at most one subtask at a time. However, any set of subtasks, including subtasks from different jobs, can be processed simultaneously, provided no two of them require the same machine. Each job has a deadline before which all of its subtasks must be completed. We would like to know whether there is a schedule in which at least $k$ jobs complete on time. Show that this scheduling problem is $NP$-complete.
3. Suppose you are following the changes in price of a stock over $n$ days, and you are representing this as a list of $n$ integers $a_1, \ldots, a_n$, some positive from when the stock went up and some negative from when it went down.

You would like to partition this list into some number of periods. Each period should be a contiguous set of days over which, in total, the stock went up about as much as it went down. One way to assess this quantitatively is as follows: given a period consisting of the values $a_i, a_{i+1}, \ldots, a_j$, we define the fluctuation of the period to be $(a_i + a_{i+1} + \cdots + a_j)^2$. Thus, if the total increase over the period equals the total decrease, the fluctuation is 0, and if the total increase is close to the total decrease, the fluctuation will be close to 0.

The net fluctuation of a partition into periods is simply the sum of the fluctuations of each period. Give an algorithm that takes a list of $n$ numbers and computes a partition whose net fluctuation is as small as possible. Your algorithm should run in time polynomial in $n$. You should give a proof of correctness and a brief analysis of the running time.

Example: Consider the list 4, -3, -4, 7, -2. The partition with the smallest net fluctuation consists of two periods [4, -3] and [-4, 7, -2]. The net fluctuation is

$$(4 + (-3))^2 + ((-4) + 7 + (-2))^2 = 1^2 + 1^2 = 2.$$

END OF EXAM