You have 120 minutes to answer all five questions.
Write your answers in the separate answer booklet.
Please turn in your question sheet and your cheat sheet with your answers.

1. You and your eight-year-old nephew Elmo decide to play a simple card game. At the beginning of the game, several cards are dealt face up in a long row. Then you and Elmo take turns removing either the leftmost or rightmost card from the row, until all the cards are gone. Each card is worth a different number of points. The player that collects the most points wins the game.

   Like most eight-year-olds who haven't studied algorithms, Elmo follows the obvious greedy strategy every time he plays: ***Elmo always takes the card with the higher point value.*** Your task is to find a strategy that will beat Elmo whenever possible. (It might seem mean to beat up on a little kid like this, but Elmo absolutely *hates* it when grown-ups let him win.)

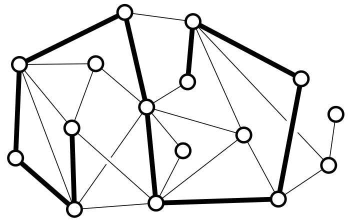
   (a) Describe an initial sequence of cards that allows you to win against Elmo, no matter who moves first, but *only* if you do *not* follow Elmo's greedy strategy.

   (b) Describe and analyze an algorithm to determine, given the initial sequence of cards, the maximum number of points that you can collect playing against Elmo.

   Here is a sample game, where both you and Elmo are using the greedy strategy. Elmo wins 8–7. You cannot win this particular game, no matter what strategy you use.

Initial cards	2	4	5	1	3
Elmo takes the 3	2	4	5	1	~~3~~
You take the 2	~~2~~	4	5	1	
Elmo takes the 4		~~4~~	5	1	
You take the 5			~~5~~	1	
Elmo takes the 1				~~1~~	

2. ***Prove*** that the following problem is NP-hard: Given an undirected graph G, find the longest path in G whose length is a multiple of 5.



This graph has a path of length 10, but no path of length 15.

3. Suppose you are given an array $A[1\,..\,n]$ of integers. Describe and analyze an algorithm that finds the largest sum of of elements in a contiguous subarray $A[i\,..\,j]$.

   For example, if the array A contains the numbers $[-6, 12, -7, 0, 14, -7, 5]$, your algorithm should return the number 19:

$$\boxed{-6}\underbrace{\boxed{12}\boxed{-7}\boxed{0}\boxed{14}}_{19}\boxed{-7}\boxed{5}$$

4. A *shuffle* of two strings X and Y is formed by interspersing the characters into a new string, keeping the characters of X and Y in the same order. For example, 'bananaananas' is a shuffle of 'banana' and 'ananas' in several different ways:

$$\texttt{banana}_{\texttt{ananas}} \qquad \texttt{ban}_{\texttt{ana}}{}^{\texttt{ana}}{}_{\texttt{nas}} \qquad {}^{\texttt{b}}{}_{\texttt{an}}{}^{\texttt{an}}{}_{\texttt{a}}{}^{\texttt{a}}{}_{\texttt{na}}{}^{\texttt{na}}{}_{\texttt{s}}$$

   The strings 'prodgyrnamammiincg' and 'dyprongarmammicing' are both shuffles of 'dynamic' and 'programming':

$${}_{\texttt{pro}}{}^{\texttt{d}}{}_{\texttt{g}}{}^{\texttt{y}}{}_{\texttt{r}}{}^{\texttt{nam}}{}_{\texttt{ammi}}{}^{\texttt{i}}{}_{\texttt{n}}{}^{\texttt{c}}{}_{\texttt{g}} \qquad {}^{\texttt{dy}}{}_{\texttt{pro}}{}^{\texttt{n}}{}_{\texttt{g}}{}^{\texttt{a}}{}_{\texttt{r}}{}^{\texttt{m}}{}_{\texttt{amm}}{}^{\texttt{ic}}{}_{\texttt{ing}}$$

   Given three strings $A[1\,..\,m]$, $B[1\,..\,n]$, and $C[1\,..\,m+n]$, describe and analyze an algorithm to determine whether C is a shuffle of A and B.

5. Suppose you are given two sorted arrays $A[1\,..\,m]$ and $B[1\,..\,n]$ and an integer k. Describe an algorithm to find the kth smallest element in the union of A and B in $\Theta(\log(m+n))$ time. For example, given the input

$$A[1\,..\,8] = [0, 1, 6, 9, 12, 13, 18, 20] \qquad B[1\,..\,5] = [2, 5, 8, 17, 19] \qquad k = 6$$

   your algorithm should return 8. You can assume that the arrays contain no duplicates. An algorithm that works only in the special case $n = m = k$ is worth 7 points.

   [Hint: What can you learn from comparing one element of A to one element of B?]